R script

```
########################Packages Required################################
library(foreign)
library(dplyr)
library(bootnet)
library(qgraph)
library(NetworkComparisonTest)
library(networktools)
library(mice)
library(psych)
library(readxl)


########################Data Sets Import################################

Data_set <- read_excel("Data set.xlsx")

Data_set = as.data.frame(Data_set, stringsAsFactors = FALSE)

Data_set = map_df(Data_set, as.numeric) %>% .[ - 1,]

#Replace -99 and 99 with NA (missing data)

Data_set[Data_set == -99] = NA

Data_set[Data_set == 99] = NA

#Select adolescents younger that 13 years or 13 years old.

Data_set = Data_set %>% filter(., Varsta_ani <= 13 | is.na(Varsta_ani))

######### descriptive statistics

table(Data_set$Gen)
```

```
table(Data_set$Clasa)

table(Data_set$Etnie)

table(Data_set$Rural_urban)

describe(Data_set$Varsta_ani)

##### Revers scoring APQ involvement and positive parenting

Data_set[, c('apq_01', 'apq_04', 'apq_07', 'apq_09', 'apq_11', 'apq_14', 'apq_15', 'apq_20', 'apq_23', 'apq_26',
             'apq_02', 'apq_05', 'apq_13', 'apq_16', 'apq_18', 'apq_27')] = 6 - Data_set[, c('apq_01', 'apq_04', 'apq_07', 'apq_09', 'apq_11', 'apq_14', 'apq_15', 'apq_20', 'apq_23', 'apq_26',
'apq_02', 'apq_05', 'apq_13', 'apq_16', 'apq_18', 'apq_27')]


########################Data Screening#################################

#### The APQ Items for the Five Parenting Domains ####


APQ <- subset(Data_set, select = c('Gen', "apq_01", "apq_02", "apq_03", "apq_04", "apq_05", "apq_06", "apq_07", "apq_08",
                                   "apq_09", "apq_10", "apq_11", "apq_12", "apq_13", "apq_14", "apq_15",
                                   "apq_16", "apq_17", "apq_18", "apq_19", "apq_20","apq_21","apq_22",
"apq_23","apq_24","apq_25","apq_26","apq_27","apq_28","apq_29","apq_30",
                                   "apq_31","apq_32","apq_33","apq_35", "apq_38"))


#Missing data APQ
```

```
table(is.na(APQ))

missing_percent = function(x){sum(is.na(x))/ length(x)*100}

apply(APQ, 2, missing_percent)


#Multivariate assumptions test

##Linearity assumptions test

random = rchisq(nrow(APQ[, -1]), 14)

fake = lm(random~., data = APQ[, -1])

standardizedResiduals = rstudent(fake)

qqnorm(standardizedResiduals)

abline(0,1)

##Normality assumtion test

describe(APQ[, -1])

hist(standardizedResiduals, breaks = 44)

describe(standardizedResiduals)



#########################Network Estimation#########################

Network <- estimateNetwork(APQ[, -c(1, 36)], default = "EBICglasso",
corMethod = "cor_auto", tuning = 0.5)

parenting_practices <- c("Parental involvement", "Positive parenting",
"Inconsistent discipline", "Parental involvement",
```

```
                         "Positive parenting", "Poor monitoring", "Parental
involvement", "Inconsistent discipline",
                         "Parental involvement", "Poor monitoring", "Parental
involvement", "Inconsistent discipline",
                         "Positive parenting", "Parental involvement",
"Parental involvement", "Positive parenting",
                         "Poor monitoring", "Positive parenting", "Poor
monitoring", "Parental involvement", "Poor monitoring",
                         "Inconsistent discipline", "Parental involvement",
"Poor monitoring", "Inconsistent discipline",
                         "Parental involvement", "Positive parenting", "Poor
monitoring", "Poor monitoring", "Poor monitoring",
                         "Inconsistent discipline", "Poor monitoring",
"Corporal punishment", "Corporal punishment")


items <- c("apq_1", "apq_2", "apq_3", "apq_4", "apq_5", "apq_6", "apq_7",
"apq_8", "apq_9", "apq_10", "apq_11", "apq_12",
           "apq_13", "apq_14", "apq_15",
"apq_16","apq_17","apq_18","apq_19","apq_20","apq_21","apq_22","apq_23","apq_
24",

"apq_25","apq_26","apq_27","apq_28","apq_29","apq_30","apq_31","apq_32","apq_
33","apq_35")


# plot and save high resolution graph


tiff("Fig. 1.tif", family = "ArialMT", units = "cm",
     width = 24, height = 16, pointsize = 12, res = 500)


plot(Network, layout = "spring", labels = items, groups =
parenting_practices)


dev.off()


####################Computing Centrality Indices##################


# plot and save high resolution graph
tiff("Fig. 2.tif", family = "ArialMT", units = "cm",
     width = 24, height = 16, pointsize = 12, res = 500)


centralityPlot(Network, include = c("Strength", "Closeness", "Betweenness"))
```

```
dev.off()

###################Edge-Weights Accuracy###########################

boot0 <- bootnet(Network, nBoots = 2500, nCores = 8)

tiff("Fig. S1.tif", family = "ArialMT", units = "cm",
     width = 30, height = 16, pointsize = 12, res = 230)
plot(boot0, labels = FALSE, order = "sample")
dev.off()

#################The Stability of Centrality Indices###################


boot2 <- bootnet(Network, nBoots = 2500, type = "case", nCores = 8,
statistics =  c("strength", "closeness", "betweenness"))
plot(boot2,  statistics =  c("strength", "closeness", "betweenness"))
corStability(boot2)

##################Significant Difference between Nodes#################

boot1 <- bootnet(Network, nBoots = 2500, nCores = 8, statistics =
c("strength", "closeness", "betweenness"))

tiff("Fig. S2.tif", family = "ArialMT", units = "cm",
     width = 30, height = 16, pointsize = 12, res = 230)
plot(boot1, labels = TRUE, order = "sample", statistics =  c("strength",
"closeness", "betweenness"))
dev.off()

###################APQ Bridge Test##################

bridge_APQ_network <- getWmat(Network)
qgraph(bridge_APQ_network, groups=parenting_practices, layout="spring")

bridge_centrality <- bridge(bridge_APQ_network,
communities=parenting_practices)
```

```
tiff("Fig. S3.tif", family = "ArialMT", units = "cm",

     width = 30, height = 16, pointsize = 12, res = 230)

plot(bridge_centrality, include=c("Bridge Strength", "Bridge Betweenness",
'Bridge Closeness',

                                  'Bridge Expected Influence (1-step)'),
order="value")

dev.off()


plot(bridge_centrality, include=c("Bridge Strength", "Bridge Betweenness",
'Bridge Closeness', 'Bridge Expected Influence (1-step)'), zscore=TRUE,
order="value")


bridge_strength <- bridge_centrality$`Bridge Strength`

top_bridges <-
names(bridge_strength[bridge_strength>quantile(bridge_strength, probs=0.80,
na.rm=TRUE)])

top_bridges

bridge_num_w1 <- which(names(bridge_strength) %in% top_bridges)

new_communities <- vector()

for(i in 1:length(bridge_strength)) {

  if(i %in% bridge_num_w1) {

    new_communities[i] <- "Bridge"

  } else {new_communities[i] <- parenting_practices[i]}

}

qgraph(bridge_APQ_network, layout="spring", legend = TRUE,
groups=new_communities,

       color = c('red', 'orange', 'gold', 'green', 'blue', 'brown'))



###############APQ Males-Females Differences Test################


APQ_males = APQ %>% filter(., Gen == 1)


APQ_females = APQ %>% filter(., Gen == 2)



Network_males <- estimateNetwork(APQ_males[, - c(1, 36)], corMethod =
'cor_auto', default = "EBICglasso")
```

```
Network_females <- estimateNetwork(APQ_females[, - c(1, 36)], corMethod =
'cor_auto', default = "EBICglasso")

plot(Network_females, layout = "spring", labels = items, groups =
parenting_practices)

plot(Network_males, layout = "spring", labels = items, groups =
parenting_practices)


Females_males_APQ_comparison <- NCT(Network_males, Network_females, it =
1000, binary.data=FALSE,
                                    paired=FALSE, weighted=TRUE, AND=TRUE,
abs=TRUE,  test.edges=TRUE,
                                    edges = 'all',
make.positive.definite=TRUE, p.adjust.methods = "BH", test.centrality=TRUE,
                                    centrality=
c("strength","expectedInfluence"), nodes="all",
communities=parenting_practices,
                                    useCommunities="all", verbose = TRUE)


plot(Females_males_APQ_comparison, what = 'network')

plot(Females_males_APQ_comparison, what = 'centrality')

plot(Females_males_APQ_comparison, what = 'edge')

plot(Females_males_APQ_comparison, what = 'strength')

print(Females_males_APQ_comparison)



centralityPlot(list(Males = Network_males, Females = Network_females),
include = c("Strength", "Closeness", "Betweenness", "expectedInfluence"),
labels = items)


#########Screening APQ_YSR Subscales Data#########################

##Compute Total Scores on Sub scales


APQ_YSR <- Data_set %>% mutate(Involvement = apq_01 + apq_04 + apq_07 +
apq_09 + apq_11 + apq_14 + apq_15 + apq_20 + apq_23 + apq_26)%>%

  mutate(Positive_Parenting = apq_02 + apq_05 + apq_13 + apq_16 + apq_18 +
apq_27) %>%

  mutate(Poor_Monitoring = apq_06 + apq_10 + apq_17 + apq_19 + apq_21 +
apq_24 + apq_28 + apq_29 + apq_30 + apq_32)%>%

  mutate(Inconsistent_Discipline = apq_03 + apq_08 + apq_12 + apq_22 + apq_25
+ apq_31) %>%

  mutate(Corporal_Punishment = apq_33 + apq_35 + apq_38) %>%

  mutate(Affective_Problems = ysr_05 + ysr_14 + ysr_18 + ysr_24 + ysr_35 +
ysr_52 + ysr_54 + ysr_76 + ysr_77 +
```

```
           ysr_91 + ysr_100 + ysr_102 + ysr_103) %>%

  mutate(Anxiety_Problems = ysr_11 + ysr_29 + ysr_30 + ysr_45 + ysr_50 +
ysr_112) %>%

  mutate(Somatic_Problems = ysr_56a + ysr_56b + ysr_56c + ysr_56d + ysr_56e +
ysr_56f + ysr_56g) %>%

  mutate(Attention_Deficit_Hyperactivity_Problems = ysr_04 + ysr_08 + ysr_10
+ ysr_41 + ysr_78 + ysr_93 + ysr_104) %>%

  mutate(Oppositional_Defiant_Problems = ysr_03 + ysr_22 + ysr_23 + ysr_86 +
ysr_95) %>%

  mutate(Conduct_Problems = ysr_16 + ysr_21 + ysr_26 + ysr_28 + ysr_37 +
ysr_39 + ysr_43 + ysr_57 + ysr_67 +

           ysr_72 + ysr_81 + ysr_82 + ysr_90 + ysr_97 + ysr_101) %>%

  select(Varsta_ani, Gen, Involvement, Positive_Parenting, Poor_Monitoring,
Inconsistent_Discipline, Corporal_Punishment, Affective_Problems,
Anxiety_Problems, Somatic_Problems,

         Attention_Deficit_Hyperactivity_Problems,
Oppositional_Defiant_Problems, Conduct_Problems)


# Reliability


#parental involvement

APQ[, c('apq_01', 'apq_04', 'apq_07', 'apq_09', 'apq_11', 'apq_14', 'apq_15',
'apq_20', 'apq_23', 'apq_26')] %>% polychoric() %>% .$rho %>% alpha() %>%
.$total


#positive parenting

APQ[, c('apq_02', 'apq_05', 'apq_13', 'apq_16', 'apq_18', 'apq_27')] %>%
polychoric() %>% .$rho %>% alpha() %>% .$total


#poor monitoring

APQ[, c('apq_06', 'apq_10', 'apq_17', 'apq_19', 'apq_21', 'apq_24', 'apq_28',
'apq_29', 'apq_30', 'apq_32')] %>% polychoric() %>% .$rho %>% alpha() %>%
.$total


#inconsistent discipline

APQ[,  c('apq_03', 'apq_08', 'apq_12', 'apq_22', 'apq_25', 'apq_31')] %>%
polychoric() %>% .$rho %>% alpha() %>% .$total


#corporal punishment

APQ[, c('apq_33', 'apq_35', 'apq_38')] %>% polychoric() %>% .$rho %>% alpha()
%>% .$total


# Affective problems
```

```
Data_set[, c('ysr_05', 'ysr_14', 'ysr_18', 'ysr_24', 'ysr_35', 'ysr_52',
'ysr_54', 'ysr_76', 'ysr_77', 'ysr_91', 'ysr_100', 'ysr_102', 'ysr_103')] %>%
polychoric() %>% .$rho %>% alpha() %>% .$total


# Anxiety problems

Data_set[, c('ysr_11', 'ysr_29', 'ysr_30', 'ysr_45', 'ysr_50', 'ysr_112')]
%>% polychoric() %>% .$rho %>% alpha() %>% .$total


# Somatic_Problems

Data_set[, c('ysr_56a', 'ysr_56b', 'ysr_56c', 'ysr_56d', 'ysr_56e',
'ysr_56f', 'ysr_56g')] %>% polychoric() %>% .$rho %>% alpha() %>% .$total


# Attention_Deficit_Hyperactivity_Problems

Data_set[, c('ysr_04', 'ysr_08', 'ysr_10', 'ysr_41', 'ysr_78', 'ysr_93',
'ysr_104')] %>% polychoric() %>% .$rho %>% alpha() %>% .$total


#Oppositional_Defiant_Problems

Data_set[, c('ysr_03', 'ysr_22', 'ysr_23', 'ysr_86', 'ysr_95')] %>%
polychoric() %>% .$rho %>% alpha() %>% .$total


#Conduct_Problems

Data_set[, c('ysr_16', 'ysr_21', 'ysr_26', 'ysr_28', 'ysr_37', 'ysr_39',
'ysr_43', 'ysr_57', 'ysr_67', 'ysr_72', 'ysr_81', 'ysr_82', 'ysr_90',
'ysr_97', 'ysr_101')] %>% polychoric() %>% .$rho %>% alpha() %>% .$total


##Missing data APQ_YSR

table(is.na(APQ_YSR))


random2 = rchisq(nrow(APQ_YSR[, -c(1,2)]), 14)

fake2 = lm(random2~., data = APQ_YSR[, -c(1,2)])

standardizedResiduals2 = rstudent(fake2)

qqnorm(standardizedResiduals2)

abline(0,1)


##Normality assumtion test


describe(APQ_YSR)

hist(standardizedResiduals2, breaks = 44)

describe(standardizedResiduals2)
```

```
#########################Multiple imputation APQ_YSR ##############

table(is.na(APQ_YSR))
imputed_APQ_YSR <- mice(APQ_YSR, m=50, maxit = 50, method = 'pmm', seed = 4014)
summary(imputed_APQ_YSR)
imputed_APQ_YSR$imp$Positive_Parenting
APQ_YSR_Complete <- complete(imputed_APQ_YSR)
which(is.na(APQ_YSR$Gen) == T)
APQ_YSR_Complete = APQ_YSR_Complete[-c(947, 950),]



###################APQ_YSR Network Estimation##################

parenting_practices_Children_m_h <- c('a)Parental involvement', 'b)Positive parenting', 'c)Poor monitoring', 'd)Inconsistent discipline',
                                      'e)Corporal punishment', 'f)Affective problems', 'g)Anxiety problems', 'h)Somatic problems',
                                      'i)ADHD problems', 'j)OD Problems', 'k)Conduct Problems')



parenting_practices_Children_m_h2 <- c("I", "PP", "PM", "ID", "CP", "Affec_P", "AP", "SP", "ADHDP", "ODP", "Con_P")


Network_APQ_YSR <- estimateNetwork(APQ_YSR_Complete[, -c(1,2)], default = "huge", tuning = 0.5)


# plot and save high resolution graph
tiff("Fig. 3.tif", family = "ArialMT", units = "cm",
     width = 24, height = 16, pointsize = 12, res = 500)
plot(Network_APQ_YSR, layout = "spring", labels = parenting_practices_Children_m_h2, groups = parenting_practices_Children_m_h,
     color = c('#58D3F7', '#0040FF', '#9F81F7', '#A9E2F3', '#F7819F', '#00FF80', '#01DF01', '#9AFE2E', '#F7FE2E', '#FAAC58', '#DC143C'))
dev.off()


#####################APQ_YSR centrality indices computation###############
tiff("Fig. S4.tif", family = "ArialMT", units = "cm",
     width = 30, height = 16, pointsize = 12, res = 230)
```

```
centralityPlot(Network_APQ_YSR, include = c("Strength", "Closeness", 
"Betweenness"), labels = parenting_practices_Children_m_h)

dev.off()


#####################APQ_YSR edge weight accuracy########################


boot0_APQ_YSR <- bootnet(Network_APQ_YSR, nBoots = 2500, nCores = 8)


tiff("Fig. S5.tif", family = "ArialMT", units = "cm",
     width = 30, height = 16, pointsize = 12, res = 230)
plot(boot0_APQ_YSR, labels = FALSE, order = "sample")
dev.off()
###########The Stability of APQ_YSR Centrality Indices stability##########



boot2_APQ_YSR <- bootnet(Network_APQ_YSR, nBoots = 2500, type = "case", 
nCores = 8,
                         statistics =  c("strength", "closeness", 
"betweenness"))


plot(boot2_APQ_YSR,  statistics =  c("strength", "closeness", "betweenness"))


corStability(boot2_APQ_YSR)


#######APQ_YSR Significant Differences (Edges, Centrality Indices) Test####


boot1_APQ_YSR <- bootnet(Network_APQ_YSR, nBoots = 2500, nCores = 8, 
statistics =  c("strength", "closeness", "betweenness"))


tiff("Fig. S6.tif", family = "ArialMT", units = "cm",
     width = 30, height = 16, pointsize = 12, res = 230)
plot(boot1_APQ_YSR, labels = TRUE, order = "sample", statistics = 
c("strength", "closeness", "betweenness"))
dev.off()


tiff("Fig. S7.tif", family = "ArialMT", units = "cm",
     width = 34, height = 24, pointsize = 12, res = 230)
plot(boot0_APQ_YSR, 'edge', plot = 'difference', onlyNonZero = TRUE, order = 
'sample')
```

```
dev.off()


###########APQ_YSR bridges###################

bridge_APQ_YSR_network <- getWmat(Network_APQ_YSR)


APQ_YSR_comunities <- c(rep('Parenting_Practices', 5),
rep('Child_M_Health_Prob', 6))


qgraph(bridge_APQ_YSR_network, groups=APQ_YSR_comunities, layout="spring")


bridge_centrality_APQ_YSR <- bridge(bridge_APQ_YSR_network, communities =
APQ_YSR_comunities, nodes = parenting_practices_Children_m_h)


plot(bridge_centrality_APQ_YSR, include=c("Bridge Strength", "Bridge
Betweenness", 'Bridge Expected Influence (1-step)'), order="value")


# plot and save high resolution graph

tiff("Fig. 4.tif", family = "ArialMT", units = "cm",

     width = 24, height = 16, pointsize = 12, res = 500)

plot(bridge_centrality_APQ_YSR, include=c("Bridge Strength", "Bridge
Closeness", 'Bridge Expected Influence (1-step)'), zscore=TRUE)

dev.off()


bridge_strength_APQ_YSR <- bridge_centrality_APQ_YSR$`Bridge Strength`


top_bridges_APQ_YSR <-
names(bridge_strength_APQ_YSR[bridge_strength_APQ_YSR>quantile(bridge_strengt
h_APQ_YSR, probs=0.80, na.rm=TRUE)])


top_bridges_APQ_YSR


bridge_num_w1_APQ_YSR <- which(names(bridge_strength_APQ_YSR) %in%
top_bridges_APQ_YSR)

new_communities_APQ_YSR <- vector()

for(s in 1:length(bridge_strength_APQ_YSR)) {

  if(s %in% bridge_num_w1_APQ_YSR) {

    new_communities_APQ_YSR[s] <- "Bridge"

  } else {new_communities_APQ_YSR[s] <- APQ_YSR_comunities[s]}

}
```

```
qgraph(bridge_APQ_YSR_network, layout="spring", legend = TRUE,
groups=new_communities_APQ_YSR, color =
         c('#F78181','#40FF00', '#2E9AFE'))


############Difference Males-Females Test#############


APQ_YSR_males <- APQ_YSR_Complete %>% filter(., Gen == 1)


APQ_YSR_females <- APQ_YSR_Complete %>% filter(., Gen == 2)


Network_males_APQ_YSR <- estimateNetwork(APQ_YSR_males[, -c(1,2)], default =
"huge", tuning = 0.5)


Network_females_APQ_YSR <- estimateNetwork(APQ_YSR_females[, -c(1,2)],
default = "huge", tuning = 0.5)


plot(Network_females_APQ_YSR, layout = "spring", labels =
parenting_practices_Children_m_h2, groups = parenting_practices_Children_m_h,
     color = c('#58D3F7', '#0040FF', '#9F81F7', '#A9E2F3', '#F7819F',
'#00FF80', '#01DF01', '#9AFE2E', '#F7FE2E',
               '#FAAC58', '#F78181'))


plot(Network_males_APQ_YSR, layout = "spring", labels =
parenting_practices_Children_m_h2, groups = parenting_practices_Children_m_h,
     color = c('#58D3F7', '#0040FF', '#9F81F7', '#A9E2F3', '#F7819F',
'#00FF80', '#01DF01', '#9AFE2E', '#F7FE2E',
               '#FAAC58', '#F78181'))


Females_males_APQ_YSR_comparison <- NCT(Network_males_APQ_YSR,
Network_females_APQ_YSR, it = 1000, binary.data=FALSE,
                                        paired=FALSE, weighted=TRUE,
AND=TRUE, abs=TRUE, test.edges=TRUE, edges = 'all',
                                        make.positive.definite=TRUE,
p.adjust.methods = "BH" , test.centrality=TRUE, centrality= c("strength",
"closeness"),
                                        nodes="all",
communities=parenting_practices_Children_m_h, useCommunities="all", verbose =
TRUE)


plot(Females_males_APQ_YSR_comparison, what = 'network')
```

```
plot(Females_males_APQ_YSR_comparison, what = 'centrality')


plot(Females_males_APQ_YSR_comparison, what = 'edge')


plot(Females_males_APQ_YSR_comparison, what = 'strength')


print(Females_males_APQ_YSR_comparison)


centralityPlot(list(Males = Network_males_APQ_YSR, Females =
Network_females_APQ_YSR), include = c("Strength", "Closeness"), labels =
parenting_practices_Children_m_h)


################ Welch t-test for independent samples APQ ######


APQ_t_test = APQ_YSR %>% mutate(APQ_total = Involvement + Positive_Parenting
+ Inconsistent_Discipline + Corporal_Punishment + Poor_Monitoring)


describe(APQ_t_test$APQ_total)


t.test(APQ_total ~ Gen, data = APQ_t_test,  var.equal = F)


cor.plot(APQ[, -c(1, 36)])
#####################END (NOT RUN)#####################
```