SINM R Script

MS

24/09/2018

```r
library(formatR)
library(foreign)
library(IsingFit)
library(qgraph)
library(igraph)
```

```
##
## Attaching package: 'igraph'
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
## The following object is masked from 'package:base':
##
##     union
```

```r
library(bootnet)
```

```
## Loading required package: ggplot2
## This is bootnet 1.0.1
## For questions and issues, please see github.com/SachaEpskamp/bootnet.
```

```r
library(IsingSampler)
```

```
## Loading required package: Rcpp
```

```r
library(compute.es)
library(NetworkComparisonTest)
library(mgm)
```

```
## This is mgm 1.2-2
## Note that the syntax changed considerably from version 1.1-8 to 1.2-0.
## Please report any bugs: https://github.com/jmbh/mgm/issues
```

```r
library(rmarkdown)
library(plyr)
```

NL

NL data preparation

```r
## load whole data set:
#dataset <- read.spss("CY6_MS_CMB_STU_QQQ.sav", use.value.labels = TRUE,
#                     to.data.frame = TRUE)
#NL_complete <- dataset[dataset$CNT=="Netherlands",] # only Dutch students
```

```r
#save(NL_complete, file = "NL_complete.Rda")

# getting data ready
load("NL_complete.Rda")

# Recode variables - creating data set per interest-related construct
# Student ID: CNTSTUID, Gender: ST004D01T
dataDemo <- data.frame(ID = NL_complete$CNTSTUID,
                       Gen = as.numeric(
                         as.character(revalue(NL_complete$ST004D01T,
                         c("Male" = "0", "Female" = "1")))))
# enjoyment items:ST094Q01NA-ST094Q05NA
dataEnj <- data.frame(Efu = as.numeric
                      (as.character(revalue(NL_complete$ST094Q01NA,
                                            c("Strongly disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly agree" = "4")))),
                      Elr = as.numeric
                      (as.character(revalue(NL_complete$ST094Q02NA,
                                            c("Strongly disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly agree" = "4")))),
                      Ehw = as.numeric
                      (as.character(revalue(NL_complete$ST094Q03NA,
                                            c("Strongly disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly agree" = "4")))),
                      Eac = as.numeric
                      (as.character(revalue(NL_complete$ST094Q04NA,
                                              c("Strongly disagree" = "1",
                                                "Disagree" = "2",
                                                "Agree" = "3",
                                                "Strongly agree" = "4")))),
                      Eil = as.numeric
                      (as.character(revalue(NL_complete$ST094Q05NA,
                                              c("Strongly disagree" = "1",
                                                "Disagree" = "2",
                                                "Agree" = "3",
                                                "Strongly agree" = "4")))))
# interest items: ST095Q04NA-ST095Q15NA
dataInt <- data.frame(Ibi = as.numeric(as.character(revalue(NL_complete$ST095Q04NA,
                                               c("Not interested" = "1",
                                                 "Hardly interested" = "2",
                                                 "Interested" = "3",
                                                 "Highly interested" = "4",
                                                 "I don't know what this is" = "NA")))),
                      Imf = as.numeric(as.character(revalue(NL_complete$ST095Q07NA,
                                               c("Not interested" = "1",
                                                 "Hardly interested" = "2",
                                                 "Interested" = "3",
```

```
                                              "Highly interested" = "4",
                                              "I don't know what this is" = "NA")))),
                       Iet = as.numeric(as.character(revalue(NL_complete$ST095Q08NA,
                                            c("Not interested" = "1",
                                              "Hardly interested" = "2",
                                              "Interested" = "3",
                                              "Highly interested" = "4",
                                              "I don't know what this is" = "NA")))),
                       Iun = as.numeric(as.character(revalue(NL_complete$ST095Q13NA,
                                            c("Not interested" = "1",
                                              "Hardly interested" = "2",
                                              "Interested" = "3",
                                              "Highly interested" = "4",
                                              "I don't know what this is" = "NA")))),
                       Ipd = as.numeric(as.character(revalue(NL_complete$ST095Q15NA,
                                            c("Not interested" = "1",
                                              "Hardly interested" = "2",
                                              "Interested" = "3",
                                              "Highly interested" = "4",
                                              "I don't know what this is" = "NA")))))
# value items: ST113Q01TA-ST113Q04TA
dataVal <- data.frame(Vwl = as.numeric(as.character(revalue(NL_complete$ST113Q01TA,
                                            c("Strongly Disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly Agree" = "4")))),
                       Vdo = as.numeric(as.character(revalue(NL_complete$ST113Q02TA,
                                            c("Strongly Disagree" = "1",
                                             "Disagree" = "2",
                                             "Agree" = "3",
                                             "Strongly Agree" = "4")))),
                       Vcp = as.numeric(as.character(revalue(NL_complete$ST113Q03TA,
                                            c("Strongly Disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly Agree" = "4")))),
                       Vhp = as.numeric(as.character(revalue(NL_complete$ST113Q04TA,
                                            c("Strongly Disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                             "Strongly Agree" = "4")))))
# behavior items: ST146Q01TA-ST146Q09NA
dataBeh <- data.frame(Btv = as.numeric(as.character(revalue(NL_complete$ST146Q01TA,
                                            c("Never or hardly ever" = "1",
                                              "Sometimes" = "2",
                                              "Regularly" = "3",
                                              "Very often" = "4")))),
                       Bbo = as.numeric(as.character(revalue(NL_complete$ST146Q02TA,
                                            c("Never or hardly ever" = "1",
                                              "Sometimes" = "2",
                                              "Regularly" = "3",
                                              "Very often" = "4")))),
                       Bws = as.numeric(as.character(revalue(NL_complete$ST146Q03TA,
```

```
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bre = as.numeric(as.character(revalue(NL_complete$ST146Q04TA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bsc = as.numeric(as.character(revalue(NL_complete$ST146Q05TA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bsn = as.numeric(as.character(revalue(NL_complete$ST146Q06NA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bst = as.numeric(as.character(revalue(NL_complete$ST146Q07NA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bwe = as.numeric(as.character(revalue(NL_complete$ST146Q08NA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))),
                      Bnb = as.numeric(as.character(revalue(NL_complete$ST146Q09NA,
                                                 c("Never or hardly ever" = "1",
                                                   "Sometimes" = "2",
                                                   "Regularly" = "3",
                                                   "Very often" = "4")))))
# science self-efficacy items
dataSel <- data.frame(Sne = as.numeric(as.character(revalue(NL_complete$ST129Q01TA,
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
                                           "I would struggle to do this on my own" = "2",
                                           "I couldn't do this" = "1")))),
                      Sea = as.numeric(as.character(revalue(NL_complete$ST129Q02TA,
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
                                           "I would struggle to do this on my own" = "2",
                                           "I couldn't do this" = "1")))),
                      Sad = as.numeric(as.character(revalue(NL_complete$ST129Q03TA,
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
                                           "I would struggle to do this on my own" = "2",
                                           "I couldn't do this" = "1")))),
                      Sdg = as.numeric(as.character(revalue(NL_complete$ST129Q04TA,
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
```

```
                                   "I would struggle to do this on my own" = "2",
                                   "I couldn't do this" = "1")))),
                  Scs = as.numeric(as.character(revalue(NL_complete$ST129Q05TA,
                                 c("I could do this easily" = "4",
                                   "I could do this with a bit of effort" = "3",
                                   "I would struggle to do this on my own" = "2",
                                   "I couldn't do this" = "1")))),
                  Slf = as.numeric(as.character(revalue(NL_complete$ST129Q06TA,
                                 c("I could do this easily" = "4",
                                   "I could do this with a bit of effort" = "3",
                                   "I would struggle to do this on my own" = "2",
                                   "I couldn't do this" = "1")))),
                  Slm = as.numeric(as.character(revalue(NL_complete$ST129Q07TA,
                                 c("I could do this easily" = "4",
                                   "I could do this with a bit of effort" = "3",
                                   "I would struggle to do this on my own" = "2",
                                   "I couldn't do this" = "1")))),
                  Sfr = as.numeric(as.character(revalue(NL_complete$ST129Q08TA,
                                 c("I could do this easily" = "4",
                                   "I could do this with a bit of effort" = "3",
                                   "I would struggle to do this on my own" = "2",
                                   "I couldn't do this" = "1")))))

# knowledge: 1 PV per subscale
dataKno <- data.frame(Kce = as.numeric(NL_complete$PV10SCEP),
                      Kcd = as.numeric(NL_complete$PV10SCED),
                      Kci = as.numeric(NL_complete$PV10SCID),
                      Kkc = as.numeric(NL_complete$PV10SKCO),
                      Kkp = as.numeric(NL_complete$PV10SKPE),
                      Ksp = as.numeric(NL_complete$PV10SSPH),
                      Ksl = as.numeric(NL_complete$PV10SSLI),
                      Kse = as.numeric(NL_complete$PV10SSES))

### SES measure - ESCS
SES <- as.numeric(NL_complete$ESCS)

### school level - PROGN
dataLev <- data.frame(Lev = as.numeric(as.character(revalue(NL_complete$PROGN,
             c("Netherlands: practical preparation for labour market" = "0",
               "Netherlands: general vocational oriented; in next year tracks into BB, KB, GL, TL" = "0
               "Netherlands: admission to ISCED3 = MBO level 1-3, post secondary, non tertiary" = "0",
               "Netherlands: admission to ISCED3 = MBO level 1-3" = "0",
               "Netherlands: admission to ISCED3 = MBO tertiary level," = "0",
               "Netherlands: Lower secondary grades of HAVO, preparing for upper secondary HAVO or MBO
               "Netherlands: Lower secondary grades of VWO, preparing for upper secondary VWO" = "2",
               "Netherlands: Upper secondary grades of HAVO, preparing for HBO (higher vocational educa
               "Netherlands: Upper secondary grades of VWO, preparing for university" = "2",
               "Netherlands: Missing" = "NA")))))
####### Age
dataAge <- data.frame(Age = as.numeric(as.character(NL_complete$AGE)))

###### Saving data
NL_networkData <- data.frame(dataDemo, dataEnj, dataInt, dataVal, dataBeh,
```

```
                             dataSel, dataKno, SES, dataLev, dataAge)
save(NL_networkData, file = "NL_networkData.Rda")
```

NL Binarizing data -> only Havo/Vwo

```
load("NL_networkData.Rda")  # all used Vs - only HAVO/VWO
NL_binary <- NL_networkData
NL_binary <- NL_binary[NL_binary$Lev == "1" | NL_binary$Lev == "2", ]  #selecting HAVO/VWO
table(NL_binary$Lev)
```

```
##
##    1    2
##  837 1656
```

```
# start binarizing Enjoyment:
NL_binary$Efu[NL_binary$Efu < 3] <- 0
NL_binary$Efu[NL_binary$Efu > 2] <- 1
NL_binary$Elr[NL_binary$Elr < 3] <- 0
NL_binary$Elr[NL_binary$Elr > 2] <- 1
NL_binary$Ehw[NL_binary$Ehw < 3] <- 0
NL_binary$Ehw[NL_binary$Ehw > 2] <- 1
NL_binary$Eac[NL_binary$Eac < 3] <- 0
NL_binary$Eac[NL_binary$Eac > 2] <- 1
NL_binary$Eil[NL_binary$Eil < 3] <- 0
NL_binary$Eil[NL_binary$Eil > 2] <- 1

# interest:
NL_binary$Ibi[NL_binary$Ibi < 3] <- 0
NL_binary$Ibi[NL_binary$Ibi > 2] <- 1
NL_binary$Imf[NL_binary$Imf < 3] <- 0
NL_binary$Imf[NL_binary$Imf > 2] <- 1
NL_binary$Iet[NL_binary$Iet < 3] <- 0
NL_binary$Iet[NL_binary$Iet > 2] <- 1
NL_binary$Iun[NL_binary$Iun < 3] <- 0
NL_binary$Iun[NL_binary$Iun > 2] <- 1
NL_binary$Ipd[NL_binary$Ipd < 3] <- 0
NL_binary$Ipd[NL_binary$Ipd > 2] <- 1

# Value:
NL_binary$Vwl[NL_binary$Vwl < 3] <- 0
NL_binary$Vwl[NL_binary$Vwl > 2] <- 1
NL_binary$Vdo[NL_binary$Vdo < 3] <- 0
NL_binary$Vdo[NL_binary$Vdo > 2] <- 1
NL_binary$Vcp[NL_binary$Vcp < 3] <- 0
NL_binary$Vcp[NL_binary$Vcp > 2] <- 1
NL_binary$Vhp[NL_binary$Vhp < 3] <- 0
NL_binary$Vhp[NL_binary$Vhp > 2] <- 1

# Behavior: We are looking at NO behavior vs. Some behavior because the mean
# is between 1-2
NL_binary$Btv[NL_binary$Btv < 2] <- 0
NL_binary$Btv[NL_binary$Btv > 1] <- 1
NL_binary$Bbo[NL_binary$Bbo < 2] <- 0
```

```r
NL_binary$Bbo[NL_binary$Bbo > 1] <- 1
NL_binary$Bws[NL_binary$Bws < 2] <- 0
NL_binary$Bws[NL_binary$Bws > 1] <- 1
NL_binary$Bre[NL_binary$Bre < 2] <- 0
NL_binary$Bre[NL_binary$Bre > 1] <- 1
NL_binary$Bsc[NL_binary$Bsc < 2] <- 0
NL_binary$Bsc[NL_binary$Bsc > 1] <- 1
NL_binary$Bsn[NL_binary$Bsn < 2] <- 0
NL_binary$Bsn[NL_binary$Bsn > 1] <- 1
NL_binary$Bst[NL_binary$Bst < 2] <- 0
NL_binary$Bst[NL_binary$Bst > 1] <- 1
NL_binary$Bwe[NL_binary$Bwe < 2] <- 0
NL_binary$Bwe[NL_binary$Bwe > 1] <- 1
NL_binary$Bnb[NL_binary$Bnb < 2] <- 0
NL_binary$Bnb[NL_binary$Bnb > 1] <- 1

# Self efficacy
NL_binary$Sne[NL_binary$Sne < 3] <- 0
NL_binary$Sne[NL_binary$Sne > 2] <- 1
NL_binary$Sea[NL_binary$Sea < 3] <- 0
NL_binary$Sea[NL_binary$Sea > 2] <- 1
NL_binary$Sad[NL_binary$Sad < 3] <- 0
NL_binary$Sad[NL_binary$Sad > 2] <- 1
NL_binary$Sdg[NL_binary$Sdg < 3] <- 0
NL_binary$Sdg[NL_binary$Sdg > 2] <- 1
NL_binary$Scs[NL_binary$Scs < 3] <- 0
NL_binary$Scs[NL_binary$Scs > 2] <- 1
NL_binary$Slf[NL_binary$Slf < 3] <- 0
NL_binary$Slf[NL_binary$Slf > 2] <- 1
NL_binary$Slm[NL_binary$Slm < 3] <- 0
NL_binary$Slm[NL_binary$Slm > 2] <- 1
NL_binary$Sfr[NL_binary$Sfr < 3] <- 0
NL_binary$Sfr[NL_binary$Sfr > 2] <- 1

# Knowledge -> Median
NL_binary$Kce[NL_binary$Kce <= median(NL_binary$Kce)] <- 0
NL_binary$Kce[NL_binary$Kce > median(NL_binary$Kce)] <- 1
NL_binary$Kcd[NL_binary$Kcd <= median(NL_binary$Kcd)] <- 0
NL_binary$Kcd[NL_binary$Kcd > median(NL_binary$Kcd)] <- 1
NL_binary$Kci[NL_binary$Kci <= median(NL_binary$Kci)] <- 0
NL_binary$Kci[NL_binary$Kci > median(NL_binary$Kci)] <- 1
NL_binary$Kkc[NL_binary$Kkc <= median(NL_binary$Kkc)] <- 0
NL_binary$Kkc[NL_binary$Kkc > median(NL_binary$Kkc)] <- 1
NL_binary$Kkp[NL_binary$Kkp <= median(NL_binary$Kkp)] <- 0
NL_binary$Kkp[NL_binary$Kkp > median(NL_binary$Kkp)] <- 1
NL_binary$Ksp[NL_binary$Ksp <= median(NL_binary$Ksp)] <- 0
NL_binary$Ksp[NL_binary$Ksp > median(NL_binary$Ksp)] <- 1
NL_binary$Ksl[NL_binary$Ksl <= median(NL_binary$Ksl)] <- 0
NL_binary$Ksl[NL_binary$Ksl > median(NL_binary$Ksl)] <- 1
NL_binary$Kse[NL_binary$Kse <= median(NL_binary$Kse)] <- 0
NL_binary$Kse[NL_binary$Kse > median(NL_binary$Kse)] <- 1

# Save
```

```r
save(NL_binary, file = "NL_binary.Rda")
```

NL data inspection -> for TABLE 2

```r
load("NL_binary.Rda")
NL_binary <- na.omit(NL_binary) # deleting missing casewise

load("NL_networkData.Rda")  # all used Vs
# Combining data sets, only includeing IDs that are in HAVO/VWO data set:
NL_nonBinary <- merge(NL_binary, NL_networkData, by="ID")
NL_nonBinary <- NL_nonBinary[45:87] # only the non-binarized data
boys <- NL_nonBinary[NL_nonBinary$Gen.y==0,]
girls <- NL_nonBinary[NL_nonBinary$Gen.y==1,]
quantile(NL_nonBinary$SES.y)
```

```
##      0%     25%     50%     75%    100%
## -2.7698  0.0212  0.5793  0.9655  2.2382
```

```r
# low SES: within lowest 25%, high SES: within highest 25%
lowSES <- NL_nonBinary[NL_nonBinary$SES.y<=0.0212,]
highSES <- NL_nonBinary[NL_nonBinary$SES.y>=0.9655,]

save(NL_nonBinary, file = "NL_nonBinary.Rda")
load("NL_nonBinary.Rda")
# Age
mean(NL_nonBinary$Age.y)
```

```
## [1] 15.7273
```

```r
sd(NL_nonBinary$Age.y)
```

```
## [1] 0.2879591
```

```r
mean(boys$Age.y)
```

```
## [1] 15.72537
```

```r
sd(boys$Age.y)
```

```
## [1] 0.2898264
```

```r
mean(girls$Age.y)
```

```
## [1] 15.72896
```

```r
sd(girls$Age.y)
```

```
## [1] 0.2864728
```

```r
mean(lowSES$Age.y)
```

```
## [1] 15.7266
```

```r
sd(lowSES$Age.y)
```

```
## [1] 0.2834215
```

```r
mean(highSES$Age.y)
```

```
## [1] 15.72679
```

```r
sd(highSES$Age.y)
```

```
## [1] 0.2813007
```

```r
# Knowledge
mean(apply(NL_nonBinary[,c(33:40)],2,mean))
```

```
## [1] 588.1523
```

```r
mean(apply(NL_nonBinary[,c(33:40)],2,sd))
```

```
## [1] 71.28149
```

```r
mean(apply(boys[,c(33:40)],2,mean))
```

```
## [1] 598.4189
```

```r
mean(apply(boys[,c(33:40)],2,sd))
```

```
## [1] 74.48065
```

```r
mean(apply(girls[,c(33:40)],2,mean))
```

```
## [1] 579.3958
```

```r
mean(apply(girls[,c(33:40)],2,sd))
```

```
## [1] 67.08434
```

```r
mean(apply(lowSES[,c(33:40)],2,mean))
```

```
## [1] 573.584
```

```r
mean(apply(lowSES[,c(33:40)],2,sd))
```

```
## [1] 70.16297
```

```r
mean(apply(highSES[,c(33:40)],2,mean))
```

```
## [1] 605.4243
```

```r
mean(apply(highSES[,c(33:40)],2,sd))
```

```
## [1] 72.19023
```

```r
# Interest
mean(apply(NL_nonBinary[,c(7:11)],2,mean))
```

```
## [1] 2.526163
```

```r
mean(apply(NL_nonBinary[,c(7:11)],2,sd))
```

```
## [1] 0.9588554
```

```r
mean(apply(boys[,c(7:11)],2,mean))
```

```
## [1] 2.623878
```

```r
mean(apply(boys[,c(7:11)],2,sd))
```

```
## [1] 0.9537264
```

```r
mean(apply(girls[,c(7:11)],2,mean))
```

```
## [1] 2.44282
```

```r
mean(apply(girls[,c(7:11)],2,sd))
```

```
## [1] 0.9265505
```

```r
mean(apply(lowSES[,c(7:11)],2,mean))
```

```
## [1] 2.473171
```

```r
mean(apply(lowSES[,c(7:11)],2,sd))
```

```
## [1] 0.9656983
```

```r
mean(apply(highSES[,c(7:11)],2,mean))
```

```
## [1] 2.667542
```

```r
mean(apply(highSES[,c(7:11)],2,sd))
```

```
## [1] 0.9356434
```

```r
# Value
mean(apply(NL_nonBinary[,c(12:15)],2,mean))
```

```
## [1] 2.534171
```

```r
mean(apply(NL_nonBinary[,c(12:15)],2,sd))
```

```
## [1] 0.9937156
```

```r
mean(apply(boys[,c(12:15)],2,mean))
```

```
## [1] 2.631378
```

```r
mean(apply(boys[,c(12:15)],2,sd))
```

```
## [1] 0.9716041
```

```r
mean(apply(girls[,c(12:15)],2,mean))
```

```
## [1] 2.451262
```

```r
mean(apply(girls[,c(12:15)],2,sd))
```

```
## [1] 1.005075
```

```r
mean(apply(lowSES[,c(12:15)],2,mean))
```

```
## [1] 2.516886
```

```r
mean(apply(lowSES[,c(12:15)],2,sd))
```

```
## [1] 0.9819809
```

```r
mean(apply(highSES[,c(12:15)],2,mean))
```

```
## [1] 2.629925
```

```r
mean(apply(highSES[,c(12:15)],2,sd))
```

```
## [1] 0.9949853
```

```r
# Enjoyment
mean(apply(NL_nonBinary[,c(2:6)],2,mean))
```

```
## [1] 2.361672
```

```r
mean(apply(NL_nonBinary[,c(2:6)],2,sd))
```

```
## [1] 0.8861035
```

```r
mean(apply(boys[,c(2:6)],2,mean))
```

```
## [1] 2.497347
```

```r
mean(apply(boys[,c(2:6)],2,sd))
```

```
## [1] 0.8952015
```

```r
mean(apply(girls[,c(2:6)],2,mean))
```

```
## [1] 2.245953
```

```r
mean(apply(girls[,c(2:6)],2,sd))
```

```
## [1] 0.8613083
```

```r
mean(apply(lowSES[,c(2:6)],2,mean))
```

```
## [1] 2.312946
```

```r
mean(apply(lowSES[,c(2:6)],2,sd))
```

```
## [1] 0.865314
```

```r
mean(apply(highSES[,c(2:6)],2,mean))
```

```
## [1] 2.464916
```

```r
mean(apply(highSES[,c(2:6)],2,sd))
```

```
## [1] 0.8941163
```

```r
# Behavior
mean(apply(NL_nonBinary[,c(16:24)],2,mean))
```

```
## [1] 1.377642
```

```r
mean(apply(NL_nonBinary[,c(16:24)],2,sd))
```

```
## [1] 0.6114127
```

```r
mean(apply(boys[,c(16:24)],2,mean))
```

```
## [1] 1.497506
```

```r
mean(apply(boys[,c(16:24)],2,sd))
```

```
## [1] 0.6913894
```

```r
mean(apply(girls[,c(16:24)],2,mean))
```

```
## [1] 1.275409
```

```r
mean(apply(girls[,c(16:24)],2,sd))
```

```
## [1] 0.5036963
```

```r
mean(apply(lowSES[,c(16:24)],2,mean))
```

```
## [1] 1.331666
```

```r
mean(apply(lowSES[,c(16:24)],2,sd))
```

```
## [1] 0.5745353
```

```r
mean(apply(highSES[,c(16:24)],2,mean))
```

```
## [1] 1.453617
```

```r
mean(apply(highSES[,c(16:24)],2,sd))
```

```
## [1] 0.66799
```

```r
# Self-Efficacy
mean(apply(NL_nonBinary[,c(25:32)],2,mean))
```

```
## [1] 2.787694
```

```r
mean(apply(NL_nonBinary[,c(25:32)],2,sd))
```

```
## [1] 0.8567465
```

```r
mean(apply(boys[,c(25:32)],2,mean))
```

```
## [1] 2.863776
```

```r
mean(apply(boys[,c(25:32)],2,sd))
```

```
## [1] 0.8333182
```

```r
mean(apply(girls[,c(25:32)],2,mean))
```

```
## [1] 2.722802
```

```r
mean(apply(girls[,c(25:32)],2,sd))
```

```
## [1] 0.8644684
```

```r
mean(apply(lowSES[,c(25:32)],2,mean))
```

```
## [1] 2.696295
```

```r
mean(apply(lowSES[,c(25:32)],2,sd))
```

```
## [1] 0.8618289
```

```r
mean(apply(highSES[,c(25:32)],2,mean))
```

```
## [1] 2.922608
```

```r
mean(apply(highSES[,c(25:32)],2,sd))
```

```
## [1] 0.8169015
```

```r
# for t-tests:
# Knowledge
mKnoA <- apply(NL_nonBinary[,c(33:40)],1,mean)
sKnoA <- apply(NL_nonBinary[,c(33:40)],1,sd)
mKnoB <- apply(boys[,c(33:40)],1,mean)
sKnoB <- apply(boys[,c(33:40)],1,sd)
mKnoG <- apply(girls[,c(33:40)],1,mean)
sKnoG <- apply(girls[,c(33:40)],1,sd)
mKnoL <- apply(lowSES[,c(33:40)],1,mean)
sKnoL <- apply(lowSES[,c(33:40)],1,sd)
```

```r
mKnoH <- apply(highSES[,c(33:40)],1,mean)
sKnoH <- apply(highSES[,c(33:40)],1,sd)

# Interest
mIntA <- apply(NL_nonBinary[,c(7:11)],1,mean)
sIntA <- apply(NL_nonBinary[,c(7:11)],1,sd)
mIntB <- apply(boys[,c(7:11)],1,mean)
sIntB <- apply(boys[,c(7:11)],1,sd)
mIntG <- apply(girls[,c(7:11)],1,mean)
sIntG <- apply(girls[,c(7:11)],1,sd)
mIntL <- apply(lowSES[,c(7:11)],1,mean)
sIntL <- apply(lowSES[,c(7:11)],1,sd)
mIntH <- apply(highSES[,c(7:11)],1,mean)
sIntH <- apply(highSES[,c(7:11)],1,sd)

# Value
mValA <- apply(NL_nonBinary[,c(12:15)],1,mean)
sValA <- apply(NL_nonBinary[,c(12:15)],1,sd)
mValB <- apply(boys[,c(12:15)],1,mean)
sValB <- apply(boys[,c(12:15)],1,sd)
mValG <- apply(girls[,c(12:15)],1,mean)
sValG <- apply(girls[,c(12:15)],1,sd)
mValL <- apply(lowSES[,c(12:15)],1,mean)
sValL <- apply(lowSES[,c(12:15)],1,sd)
mValH <- apply(highSES[,c(12:15)],1,mean)
sValH <- apply(highSES[,c(12:15)],1,sd)

# Enjoyment
mEnjA <- apply(NL_nonBinary[,c(2:6)],1,mean)
sEnjA <- apply(NL_nonBinary[,c(2:6)],1,sd)
mEnjB <- apply(boys[,c(2:6)],1,mean)
sEnjB <- apply(boys[,c(2:6)],1,sd)
mEnjG <- apply(girls[,c(2:6)],1,mean)
sEnjG <- apply(girls[,c(2:6)],1,sd)
mEnjL <- apply(lowSES[,c(2:6)],1,mean)
sEnjL <- apply(lowSES[,c(2:6)],1,sd)
mEnjH <- apply(highSES[,c(2:6)],1,mean)
sEnjH <- apply(highSES[,c(2:6)],1,sd)

# Behavior
mBehA <- apply(NL_nonBinary[,c(16:24)],1,mean)
sBehA <- apply(NL_nonBinary[,c(16:24)],1,sd)
mBehB <- apply(boys[,c(16:24)],1,mean)
sBehB <- apply(boys[,c(16:24)],1,sd)
mBehG <- apply(girls[,c(16:24)],1,mean)
sBehG <- apply(girls[,c(16:24)],1,sd)
mBehL <- apply(lowSES[,c(16:24)],1,mean)
sBehL <- apply(lowSES[,c(16:24)],1,sd)
mBehH <- apply(highSES[,c(16:24)],1,mean)
sBehH <- apply(highSES[,c(16:24)],1,sd)

# Self-Efficacy
mSelA <- apply(NL_nonBinary[,c(25:32)],2,mean)
```

```r
sSelA <- apply(NL_nonBinary[,c(25:32)],2,sd)
mSelB <- apply(boys[,c(25:32)],2,mean)
sSelB <- apply(boys[,c(25:32)],2,sd)
mSelG <- apply(girls[,c(25:32)],2,mean)
sSelG <- apply(girls[,c(25:32)],2,sd)
mSelL <- apply(lowSES[,c(25:32)],2,mean)
sSelL <- apply(lowSES[,c(25:32)],2,sd)
mSelH <- apply(highSES[,c(25:32)],2,mean)
sSelH <- apply(highSES[,c(25:32)],2,sd)

# Knowledge
t.test(mKnoB, mKnoG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mKnoB and mKnoG
## t = 6.7872, df = 1975, p-value = 1.507e-11
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  13.52640 24.51994
## sample estimates:
## mean of x mean of y
##  598.4189  579.3958
```

```r
t.test(mKnoL, mKnoH, var.equal = FALSE) # perform t-test on the SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mKnoL and mKnoH
## t = -8.0557, df = 1063.5, p-value = 2.107e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -39.59599 -24.08473
## sample estimates:
## mean of x mean of y
##  573.5840  605.4243
```

```r
# Interest
t.test(mIntB, mIntG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mIntB and mIntG
## t = 6.1524, df = 2056.7, p-value = 9.15e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1233439 0.2387715
## sample estimates:
## mean of x mean of y
##  2.623878  2.442820
```

```r
t.test(mIntL, mIntH, var.equal = FALSE) # perform t-test on SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mIntL and mIntH
## t = -4.7431, df = 1062.5, p-value = 2.392e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2747826 -0.1139604
## sample estimates:
## mean of x mean of y
##  2.473171  2.667542
```

```r
# Value
t.test(mValB, mValG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mValB and mValG
## t = 4.4636, df = 2095, p-value = 8.486e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1009818 0.2592493
## sample estimates:
## mean of x mean of y
##  2.631378  2.451262
```

```r
t.test(mValL, mValH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mValL and mValH
## t = -1.9795, df = 1064, p-value = 0.04802
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2250901519 -0.0009886474
## sample estimates:
## mean of x mean of y
##  2.516886  2.629925
```

```r
# Enjoyment
t.test(mEnjB, mEnjG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mEnjB and mEnjG
## t = 7.1344, df = 2039.8, p-value = 1.345e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1822897 0.3204982
## sample estimates:
## mean of x mean of y
##  2.497347  2.245953
```

```r
t.test(mEnjL, mEnjH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mEnjL and mEnjH
## t = -3.0676, df = 1062.7, p-value = 0.002212
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.24917789 -0.05476208
## sample estimates:
## mean of x mean of y
##  2.312946  2.464916
```

```r
# Behavior
t.test(mBehB, mBehG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mBehB and mBehG
## t = 12.241, df = 1738, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1865101 0.2576841
## sample estimates:
## mean of x mean of y
##  1.497506  1.275409
```

```r
t.test(mBehL, mBehH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mBehL and mBehH
## t = -4.6167, df = 1036.8, p-value = 4.387e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.17378458 -0.07011786
## sample estimates:
## mean of x mean of y
##  1.331666  1.453617
```

```r
# Self-Efficacy
t.test(mSelB, mSelG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mSelB and mSelG
## t = 0.98121, df = 13.848, p-value = 0.3433
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1674932  0.4494393
## sample estimates:
## mean of x mean of y
```

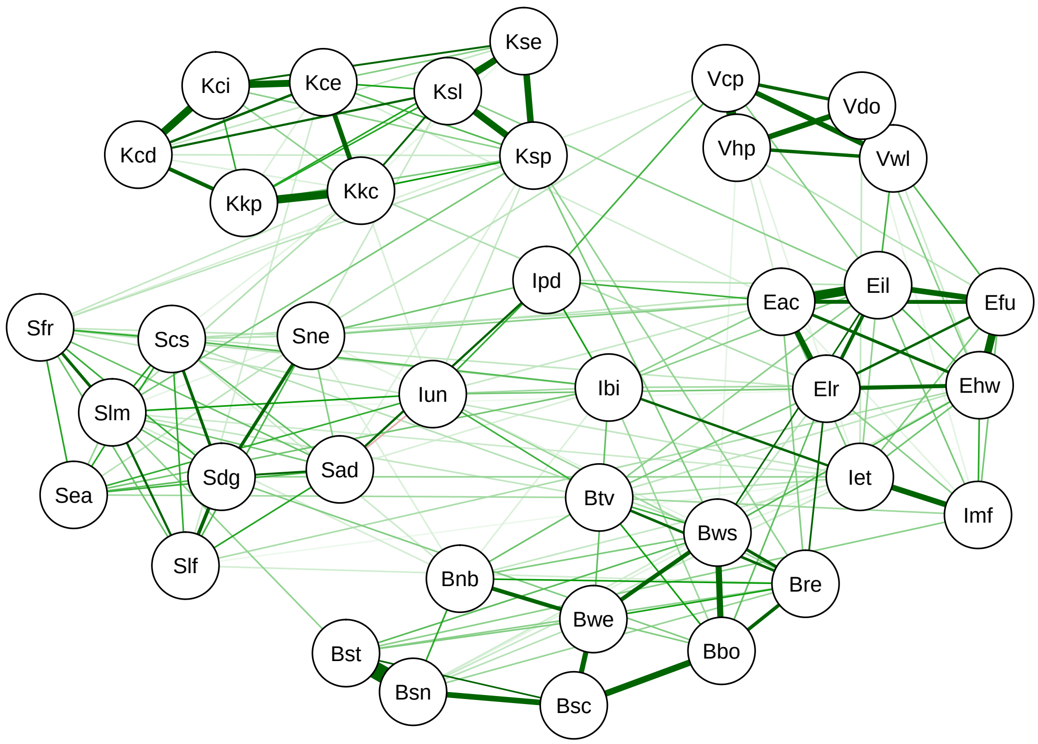
```
## 2.863776 2.722802
```

```r
t.test(mSelL, mSelH, var.equal = FALSE) # perform t-test SES difference
```

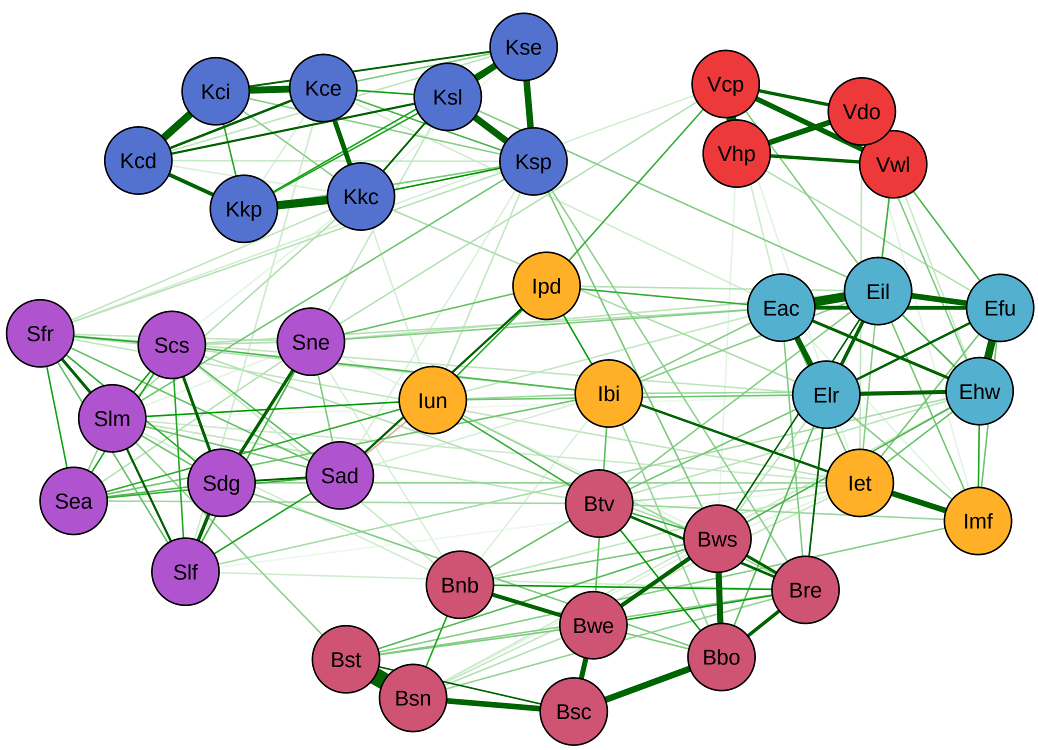
```
##
##  Welch Two Sample t-test
##
## data:  mSelL and mSelH
## t = -1.6127, df = 13.575, p-value = 0.1298
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.52817222  0.07554558
## sample estimates:
## mean of x mean of y
##  2.696295  2.922608
```

NL network estimation

```r
load("NL_binary.Rda")
NL_binary <- na.omit(NL_binary[, -c(1, 2, 42:44)])  # not in network: ID, gender, SES, Lev, Age
NL_binary <- na.omit(NL_binary)  # deleting missing casewise

# Network estimation
groups_type <- list(Enjoyment = c(1, 2, 3, 4, 5), Interest = c(6, 7, 8, 9, 10),
    Value = c(11, 12, 13, 14), Behavior = c(15, 16, 17, 18, 19, 20, 21, 22,
        23), `Self-efficacy` = c(24, 25, 26, 27, 28, 29, 30, 31), Knowledge = c(32,
        33, 34, 35, 36, 37, 38, 39))

# pick some nice colors
group_col <- c("#53B0CF", "#FFB026", "#ED3939", "#cf5372", "#b053cf", "#5372cf")

names <- c("Fun learning", "Like reading", "Happy working on", "Enjoy acquiring knowledge",
    "Interested learning about", "Biosphere", "Motion & forces", "Energy & transformations",
    "Universe", "Preventing disease", "Later work", "Need later work", "Improve career",
    "Help job", "Watch TV", "Buy books", "Websites", "Reading magazine", "Science club",
    "Simulate natural", "Simulate technical", "Website organisation", "News blogs",
    "Health issue", "Earthquakes", "Antibiotics", "Garbage disposal", "Survival Species",
    "Slf: Food labelling", "Life on Mars", "Acid rain", "Competency: Explain",
    "Competency: Evaluate", "Competency: Interpret", "Knowledge: Content", "Knowledge: Procedural & Epis
    "Systems: Physical", "Systems: Living", "Systems: Earth & Science")

# This command fits the network model:
NL_fit <- IsingFit(NL_binary)
```




```
Nl_graph <- qgraph(NL_fit$weiadj, layout = "spring", color = group_col, groups = groups_type,
    nodeNames = names, legend = FALSE, legend.mode = "style2", legend.cex = 0.3)
```



NL Community detection -> see NL vs Col for graph

```
NL_igraph <- graph_from_adjacency_matrix(abs(NL_fit$weiadj), "undirected", weighted = TRUE,
    add.colnames = FALSE)
NL_Com <- cluster_walktrap(NL_igraph)
communities(NL_Com)
```

NL Centrality -> see NL vs. Col

NL Smallworldness

NL Small worldness -> as a double check

```
smallworldness(Nl_graph, B = 1000, up = 0.995, lo = 0.005)
```

```
##       smallworldness         trans_target averagelength_target
##            1.4810141            0.4521605            1.9635628
##          trans_rnd_M         trans_rnd_lo         trans_rnd_up
##            0.2790910            0.2438194            0.3132716
##   averagelength_rnd_M averagelength_rnd_lo averagelength_rnd_up
##            1.7949703            1.7732794            1.8178205
```

NL Small worldness: get upper CI

```
# function from Dalege, used in CAN paper 2016, one change made:
# transivity calculated as in smallworldness function (took out 'average' argument)
SW_Index <- function (Graph, ci = c (.1, .05, .01, .001)) #
{
  randomC <- vector (, 1000)
  randomL <- vector (, 1000)
  for (i in 1:1000)
  {
    Rgraph <- erdos.renyi.game (vcount (Graph), ecount (Graph), 'gnm')
    randomC [i] <- transitivity (Rgraph)
    randomL [i] <- average.path.length(Rgraph)
  }
  MrandomC <- mean (randomC)
  MrandomL <- mean (randomL)
  Clustering.Graph = transitivity (Graph)
  ASPL.Graph = average.path.length (Graph)
  Index <- (Clustering.Graph / MrandomC) / (ASPL.Graph / MrandomL)

  sm_sample <- vector (, 1000)
  for (i in 1:1000)
  {
    Rgraph <- erdos.renyi.game (vcount (Graph), ecount (Graph), 'gnm')
    sm_sample [i] <- (transitivity (Rgraph) / MrandomC) /
      (average.path.length(Rgraph) / MrandomL)
  }
  CI <- as.vector (((quantile (sm_sample, 1 - (ci / 2)) -
                       quantile (sm_sample, ci / 2)) / 2) + 1)
  return (list (SW.Index = Index, Upper.CI = data.frame (CI = ci, Value.CI = CI),
                Clustering.Graph = Clustering.Graph, Clustering.Random.Graph = MrandomC,
```

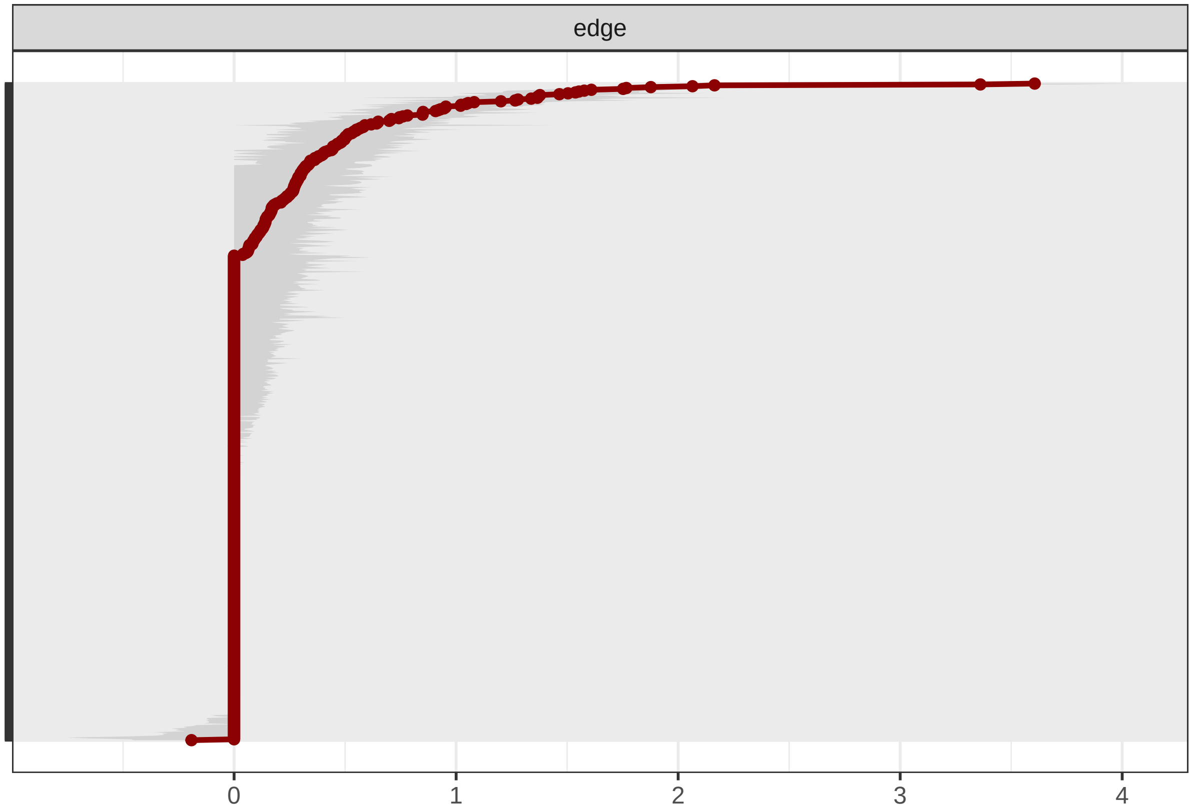
```r
                    ASPL.Graph = ASPL.Graph, ASPL.Random.Graph = MrandomL))
}

SW_Index(NL_igraph)
```

```
## $SW.Index
## [1] 1.574455
##
## $Upper.CI
##      CI Value.CI
## 1 0.100 1.093104
## 2 0.050 1.108863
## 3 0.010 1.139431
## 4 0.001 1.166091
##
## $Clustering.Graph
## [1] 0.4521605
##
## $Clustering.Random.Graph
## [1] 0.2611475
##
## $ASPL.Graph
## [1] 1.963563
##
## $ASPL.Random.Graph
## [1] 1.785535
```

NL Simulation: Connectivity -> see NL vs Col

NL Stability: edge stability

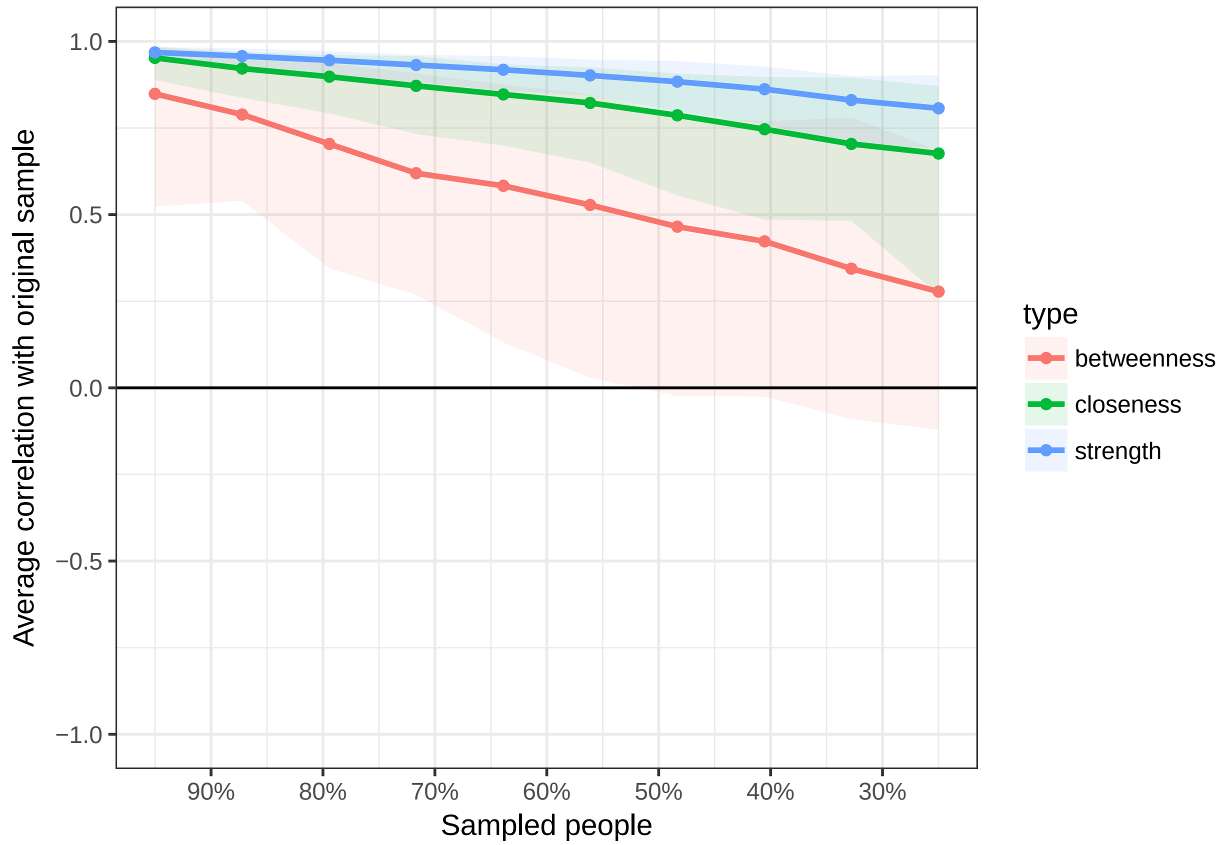
```r
#set.seed(1)
#NL_binaryStability <- bootnet(NL_binary, 1000, 'IsingFit')
#save(NL_binaryStability, file = "NL_binaryStability.Rda")
load("NL_binaryStability.Rda")

# Figure S1 A
#tiff(filename = "Figure S1 A.tiff",
#    width = 6400, height = 6400, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
plot(NL_binaryStability, order = 'sample', labels = FALSE)
```




```
#dev.off()
```

NL Stability: Centrality

```
#set.seed(1)
#NL_CentalityStability <- bootnet(NL_binary, 1000, 'IsingFit', 'person')
#save(NL_CentalityStability, file = "NL_CentalityStability.Rda")

load("NL_CentalityStability.Rda")

# Figure S1 B
#tiff(filename = "Figure S1 B.tiff",
#    width = 6400, height = 6400, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
plot(NL_CentalityStability)
```




```r
#dev.off()

corStability(NL_CentalityStability) # CS factor
```

```
## === Correlation Stability Analysis ===
##
## Sampling levels tested:
##    nPerson Drop%   n
## 1      532  75.0  96
## 2      698  67.2  82
## 3      863  59.5  97
## 4     1029  51.7 101
## 5     1195  43.9 101
## 6     1360  36.1 117
## 7     1526  28.3  94
## 8     1691  20.6 119
## 9     1857  12.8  91
## 10    2023   5.0 102
##
## Maximum drop proportions to retain correlation of 0.7 in at least 95% of the samples:
##
## betweenness: 0
##   - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.05)
##
## closeness: 0.361
##   - For more accuracy, run bootnet(..., caseMin = 0.283, caseMax = 0.439)
##
```

```
## strength: 0.75
##   - For more accuracy, run bootnet(..., caseMin = 0.672, caseMax = 1)
##
## Accuracy can also be increased by increasing both 'nBoots' and 'caseN'.
```

NL Boys vs. girls

NL Boys vs. girls Data

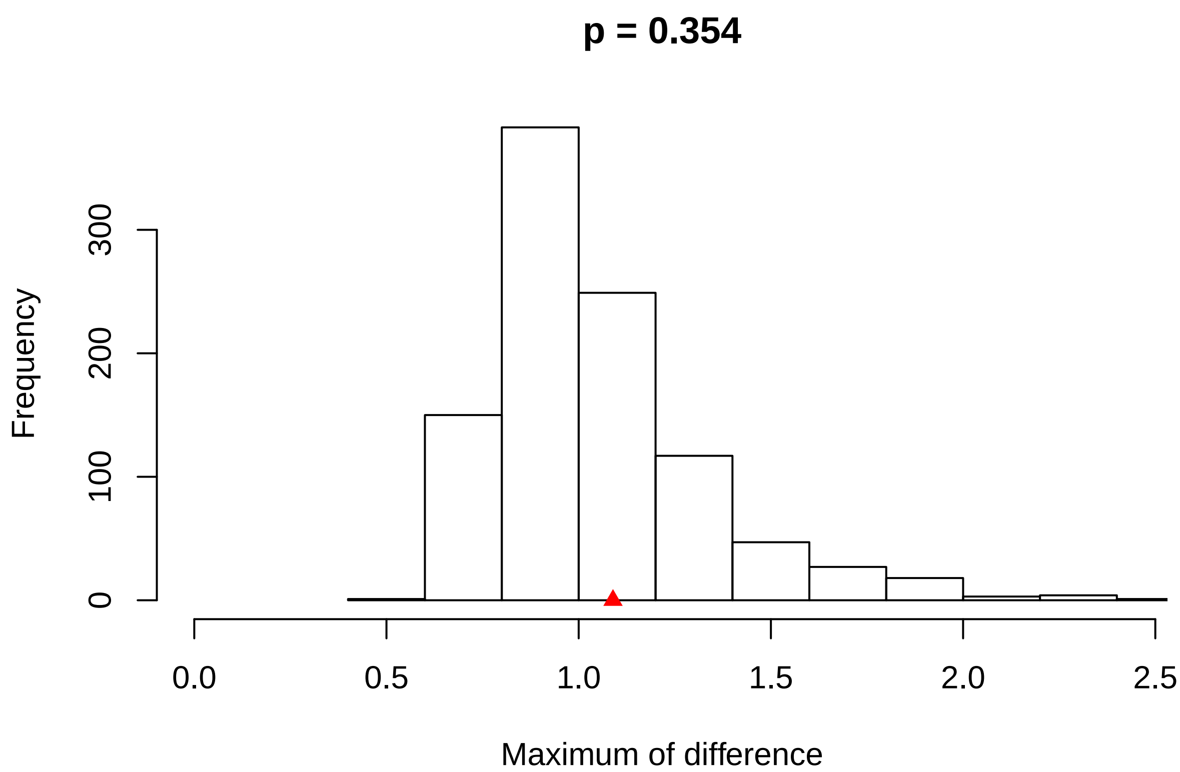
```r
load("NL_binary.Rda")
dataBinary <- NL_binary

# data frames boys vs. girls
boysBinary <- dataBinary[dataBinary$Gen=="0",]
girlsBinary <- dataBinary[dataBinary$Gen=="1",]

#
boysBinary <- na.omit(boysBinary[,-c(1,2,42:44)]) # not in network: gender, SES etc
girlsBinary <- na.omit(girlsBinary[,-c(1,2,42:44)]) # not in network: gender, SES ect
```

NL Boys vs. Girls NCT

```r
# NL_NCT_gender <- NCT(boysBinary, girlsBinary, it = 1000, gamma = .25,
# binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges = 'all')
# save(NL_NCT_gender, file='NL_NCT_gender.Rda')
load("NL_NCT_gender.Rda")
```

NL Boys vs. girls NCT -> structure invariance

```r
# network structure invariant?
plot(NL_NCT_gender, what = "network")
```




```
NL_NCT_gender$nwinv.real
```

```
## [1] 1.089276
```

```
NL_NCT_gender$nwinv.pval
```

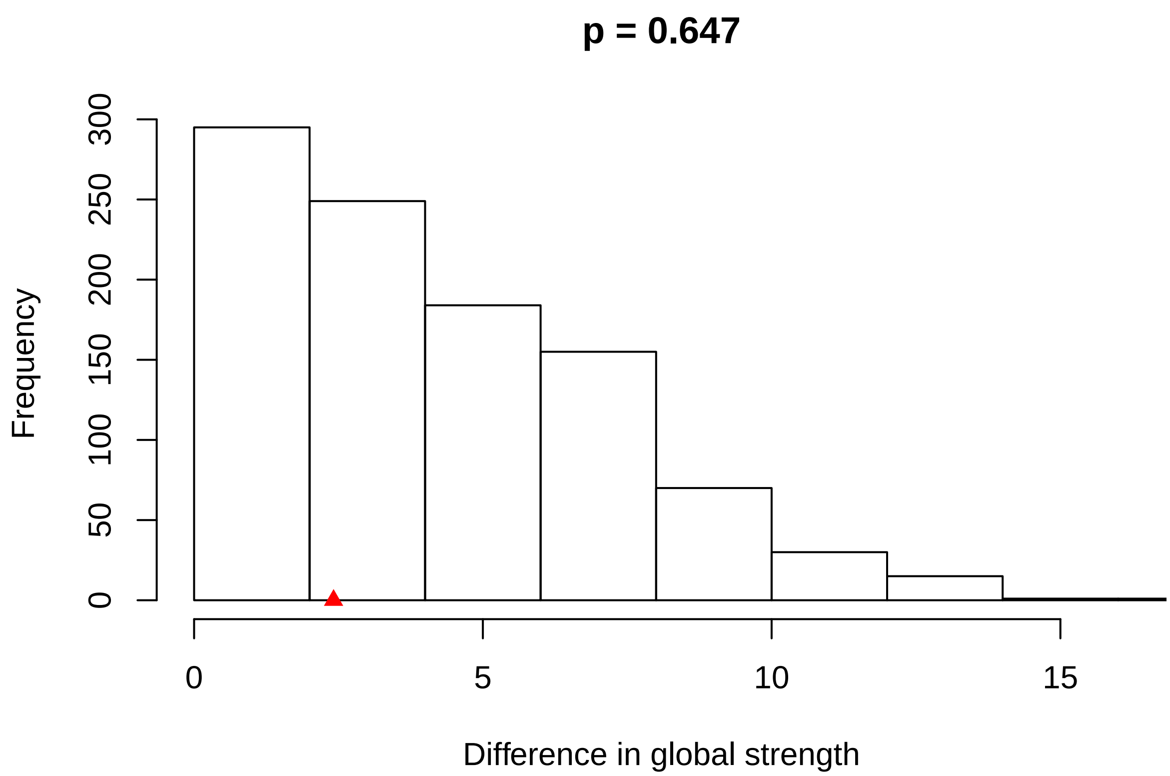
```
## [1] 0.354
```

NL Boys vs. girls NCT -> specific edges differ? -> not done because networks not invariant

```
#p.edges <- NL_NCT_gender$einv.pvals
```

NL Boys vs. girls NCT -> global strength

```
# global strength
plot(NL_NCT_gender, what = "strength")
```




```
NL_NCT_gender$glstrinv.real
```

```
## [1] 2.415292
```

```
NL_NCT_gender$glstrinv.pval
```

```
## [1] 0.647
```

```
# global strength per network
NL_NCT_gender$glstrinv.sep
```

```
## [1] 80.77602 83.19131
```

NL Low vs. high SES

NL Low vs. high SES Data

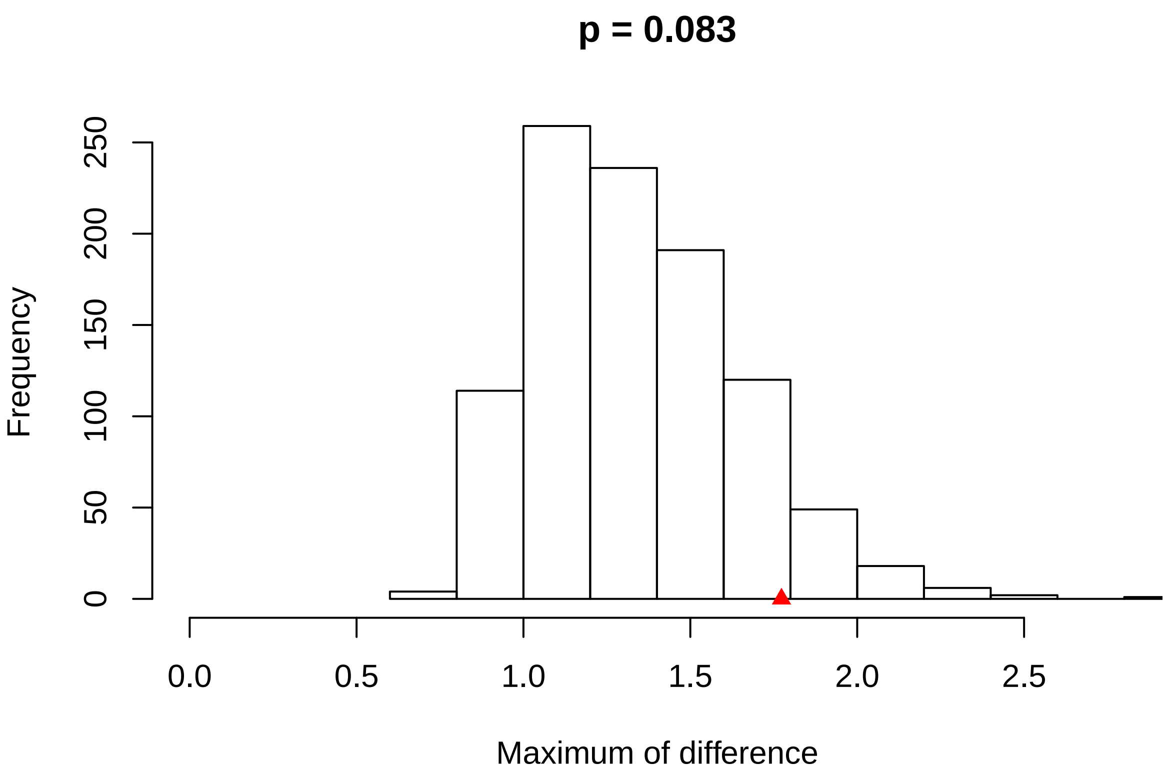
```
load("NL_binary.Rda")

# data frames low vs high SES
lowSES <- dataBinary[dataBinary$SES <= 0.0212, ]  # for quantiles: see data inspection
highSES <- dataBinary[dataBinary$SES >= 0.9655, ]

#
lowSES <- na.omit(lowSES[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age
highSES <- na.omit(highSES[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age
```

NL Low vs High SES NCT

```
#NL_NCT_SES <- NCT(lowSES, highSES, it = 1000, gamma = .25, binary.data = TRUE,
#                  paired = FALSE, test.edges = TRUE, edges = 'all')
#save(NL_NCT_SES, file="NL_NCT_SES.Rda")

load("NL_NCT_SES.Rda")
```

NL Low vs. high SES NCT -> structure invariance

```
plot(NL_NCT_SES, what = "network")
```




```
NL_NCT_SES$nwinv.real
```

```
## [1] 1.773116
```

```
NL_NCT_SES$nwinv.pval
```

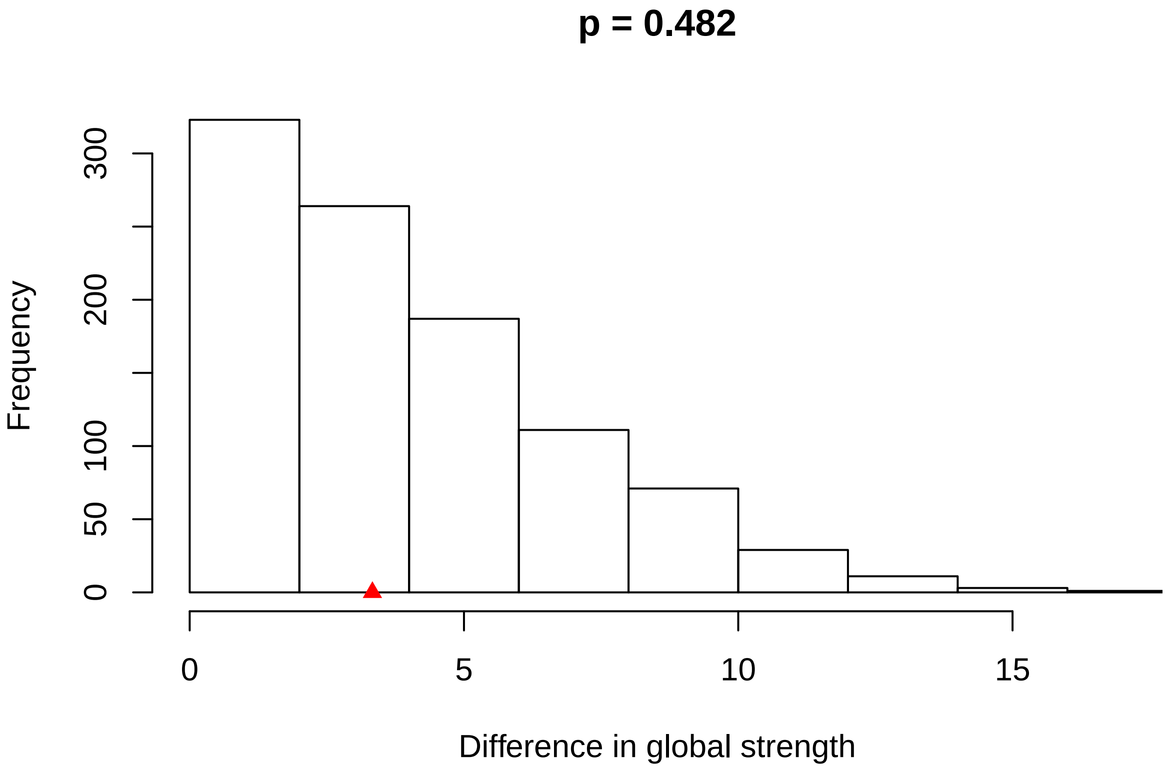
```
## [1] 0.083
```

NL Low vs. high SES NCT -> specific edges differ? -> not done because networks not invariant

```
#p.edges <- NL_NCT_SES$einv.pvals
#which(p.edges$`p-value`<0.05)
```

NL Low vs. high SES NCT -> global strength

```r
plot(NL_NCT_SES, what = "strength")
```




```r
NL_NCT_SES$glstrinv.real
```

```
## [1] 3.331138
```

```r
NL_NCT_SES$glstrinv.pval
```

```
## [1] 0.482
```

```r
# global strength per network
NL_NCT_SES$glstrinv.sep
```

```
## [1] 69.42906 66.09792
```

NL HAVO vs. VWO

NL HAVO vs. VWO Data & preparation -> create Vwo_small

```r
load("NL_binary.Rda")

HavoBinary <- NL_binary[NL_binary$Lev == "1", ]
VwoBinary <- NL_binary[NL_binary$Lev == "2", ]

HavoBinary <- na.omit(HavoBinary[, -c(1, 2, 42:44)])  # not in network: ID, gender, SES, Lev, Age,
VwoBinary <- na.omit(VwoBinary[, -c(1, 2, 42:44)])  # not in network: ID, gender, SES, Lev, Age

# small VWO sample -> same size as HAVO set.seed(1) VwoBinary_small <-
# na.omit(VwoBinary[sample(1:1441, 688,replace = FALSE),])
# save(VwoBinary_small, file = 'VwoBinary_small.Rda')
```

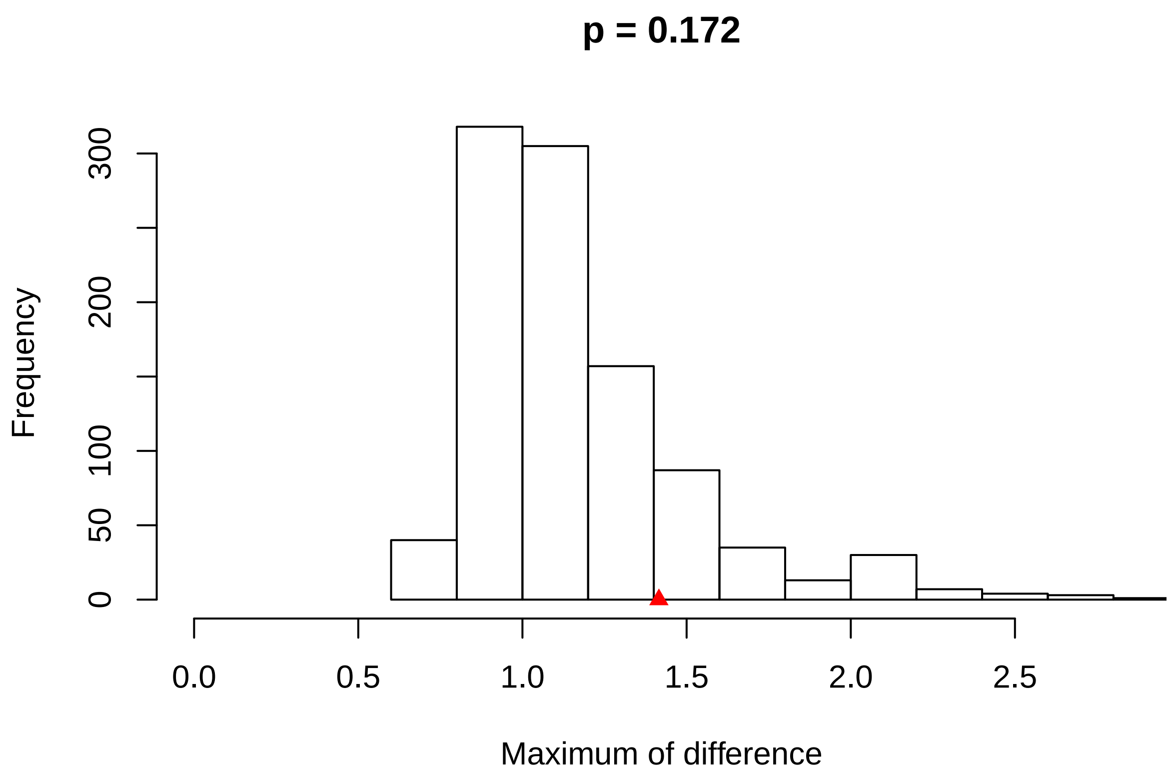
```r
load("VwoBinary_small.Rda")
```

NL Havo vs Vwo_small NCT

```r
# NL_NCT_HavoVwo_small <- NCT(HavoBinary, VwoBinary_small, it = 1000, gamma
# = .25, binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges =
# 'all') save(NL_NCT_HavoVwo_small, file='NL_NCT_HavoVwo_small.Rda')
load("NL_NCT_HavoVwo_small.Rda")
```

NL HAVO vs. VWO_small NCT -> structure invariance

```r
# network structure invariant?
plot(NL_NCT_HavoVwo_small, what = "network")
```




```r
NL_NCT_HavoVwo_small$nwinv.real
```

```
## [1] 1.415655
```

```r
NL_NCT_HavoVwo_small$nwinv.pval
```

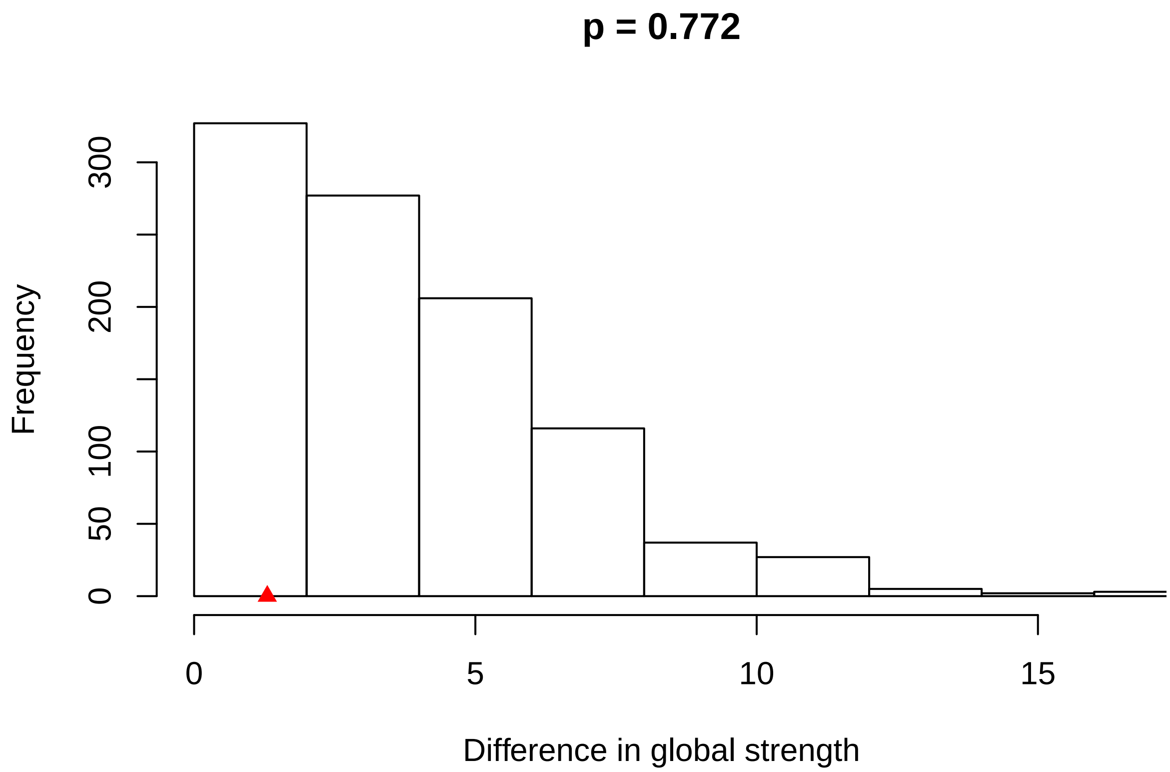
```
## [1] 0.172
```

NL HAVO vs. VWO_small NCT -> specific edges differ? -> not done because networks not invariant

```r
#p.edges <- NL_NCT_HavoVwo_small$einv.pvals #
#which(p.edges$`p-value`<0.05)
```

NL HAVO vs. VWO_small NCT -> global strength

```r
plot(NL_NCT_HavoVwo_small, what = "strength")
```




```r
NL_NCT_HavoVwo_small$glstrinv.real
```

```
## [1] 1.301064
```

```r
NL_NCT_HavoVwo_small$glstrinv.pval
```

```
## [1] 0.772
```

```r
# global strength per network
NL_NCT_HavoVwo_small$glstrinv.sep
```

```
## [1] 74.20462 72.90356
```

NL HAVO/VWO vs. MBO

NL HAVO/VWO vs. MBO Data

```r
load("NL_networkData.Rda")  # all used Vs
load("NL_binary.Rda")

# only MBO
NL_binaryMbo <- NL_networkData
NL_binaryMbo <- NL_binaryMbo[NL_binaryMbo$Lev == "0", ]

# Havo/ Vwo
NL_binaryHavoVwo <- na.omit(NL_binary[, -c(1, 2, 42:44)])
```

NL HAVO/VWO vs. MBO Data -> binarize MBO

```
# start binarizing Enjoyment:
NL_binaryMbo$Efu[NL_binaryMbo$Efu < 3] <- 0
NL_binaryMbo$Efu[NL_binaryMbo$Efu > 2] <- 1
NL_binaryMbo$Elr[NL_binaryMbo$Elr < 3] <- 0
NL_binaryMbo$Elr[NL_binaryMbo$Elr > 2] <- 1
NL_binaryMbo$Ehw[NL_binaryMbo$Ehw < 3] <- 0
NL_binaryMbo$Ehw[NL_binaryMbo$Ehw > 2] <- 1
NL_binaryMbo$Eac[NL_binaryMbo$Eac < 3] <- 0
NL_binaryMbo$Eac[NL_binaryMbo$Eac > 2] <- 1
NL_binaryMbo$Eil[NL_binaryMbo$Eil < 3] <- 0
NL_binaryMbo$Eil[NL_binaryMbo$Eil > 2] <- 1

# interest:
NL_binaryMbo$Ibi[NL_binaryMbo$Ibi < 3] <- 0
NL_binaryMbo$Ibi[NL_binaryMbo$Ibi > 2] <- 1
NL_binaryMbo$Imf[NL_binaryMbo$Imf < 3] <- 0
NL_binaryMbo$Imf[NL_binaryMbo$Imf > 2] <- 1
NL_binaryMbo$Iet[NL_binaryMbo$Iet < 3] <- 0
NL_binaryMbo$Iet[NL_binaryMbo$Iet > 2] <- 1
NL_binaryMbo$Iun[NL_binaryMbo$Iun < 3] <- 0
NL_binaryMbo$Iun[NL_binaryMbo$Iun > 2] <- 1
NL_binaryMbo$Ipd[NL_binaryMbo$Ipd < 3] <- 0
NL_binaryMbo$Ipd[NL_binaryMbo$Ipd > 2] <- 1

# Value:
NL_binaryMbo$Vwl[NL_binaryMbo$Vwl < 3] <- 0
NL_binaryMbo$Vwl[NL_binaryMbo$Vwl > 2] <- 1
NL_binaryMbo$Vdo[NL_binaryMbo$Vdo < 3] <- 0
NL_binaryMbo$Vdo[NL_binaryMbo$Vdo > 2] <- 1
NL_binaryMbo$Vcp[NL_binaryMbo$Vcp < 3] <- 0
NL_binaryMbo$Vcp[NL_binaryMbo$Vcp > 2] <- 1
NL_binaryMbo$Vhp[NL_binaryMbo$Vhp < 3] <- 0
NL_binaryMbo$Vhp[NL_binaryMbo$Vhp > 2] <- 1

# Behavior:
NL_binaryMbo$Btv[NL_binaryMbo$Btv < 2] <- 0
NL_binaryMbo$Btv[NL_binaryMbo$Btv > 1] <- 1
NL_binaryMbo$Bbo[NL_binaryMbo$Bbo < 2] <- 0
NL_binaryMbo$Bbo[NL_binaryMbo$Bbo > 1] <- 1
NL_binaryMbo$Bws[NL_binaryMbo$Bws < 2] <- 0
NL_binaryMbo$Bws[NL_binaryMbo$Bws > 1] <- 1
NL_binaryMbo$Bre[NL_binaryMbo$Bre < 2] <- 0
NL_binaryMbo$Bre[NL_binaryMbo$Bre > 1] <- 1
NL_binaryMbo$Bsc[NL_binaryMbo$Bsc < 2] <- 0
NL_binaryMbo$Bsc[NL_binaryMbo$Bsc > 1] <- 1
NL_binaryMbo$Bsn[NL_binaryMbo$Bsn < 2] <- 0
NL_binaryMbo$Bsn[NL_binaryMbo$Bsn > 1] <- 1
NL_binaryMbo$Bst[NL_binaryMbo$Bst < 2] <- 0
NL_binaryMbo$Bst[NL_binaryMbo$Bst > 1] <- 1
NL_binaryMbo$Bwe[NL_binaryMbo$Bwe < 2] <- 0
NL_binaryMbo$Bwe[NL_binaryMbo$Bwe > 1] <- 1
NL_binaryMbo$Bnb[NL_binaryMbo$Bnb < 2] <- 0
```

```r
NL_binaryMbo$Bnb[NL_binaryMbo$Bnb > 1] <- 1

# Self efficacy
NL_binaryMbo$Sne[NL_binaryMbo$Sne < 3] <- 0
NL_binaryMbo$Sne[NL_binaryMbo$Sne > 2] <- 1
NL_binaryMbo$Sea[NL_binaryMbo$Sea < 3] <- 0
NL_binaryMbo$Sea[NL_binaryMbo$Sea > 2] <- 1
NL_binaryMbo$Sad[NL_binaryMbo$Sad < 3] <- 0
NL_binaryMbo$Sad[NL_binaryMbo$Sad > 2] <- 1
NL_binaryMbo$Sdg[NL_binaryMbo$Sdg < 3] <- 0
NL_binaryMbo$Sdg[NL_binaryMbo$Sdg > 2] <- 1
NL_binaryMbo$Scs[NL_binaryMbo$Scs < 3] <- 0
NL_binaryMbo$Scs[NL_binaryMbo$Scs > 2] <- 1
NL_binaryMbo$Slf[NL_binaryMbo$Slf < 3] <- 0
NL_binaryMbo$Slf[NL_binaryMbo$Slf > 2] <- 1
NL_binaryMbo$Slm[NL_binaryMbo$Slm < 3] <- 0
NL_binaryMbo$Slm[NL_binaryMbo$Slm > 2] <- 1
NL_binaryMbo$Sfr[NL_binaryMbo$Sfr < 3] <- 0
NL_binaryMbo$Sfr[NL_binaryMbo$Sfr > 2] <- 1

# Knowledge -> Median
NL_binaryMbo$Kce[NL_binaryMbo$Kce <= median(NL_binaryMbo$Kce)] <- 0
NL_binaryMbo$Kce[NL_binaryMbo$Kce > median(NL_binaryMbo$Kce)] <- 1
NL_binaryMbo$Kcd[NL_binaryMbo$Kcd <= median(NL_binaryMbo$Kcd)] <- 0
NL_binaryMbo$Kcd[NL_binaryMbo$Kcd > median(NL_binaryMbo$Kcd)] <- 1
NL_binaryMbo$Kci[NL_binaryMbo$Kci <= median(NL_binaryMbo$Kci)] <- 0
NL_binaryMbo$Kci[NL_binaryMbo$Kci > median(NL_binaryMbo$Kci)] <- 1
NL_binaryMbo$Kkc[NL_binaryMbo$Kkc <= median(NL_binaryMbo$Kkc)] <- 0
NL_binaryMbo$Kkc[NL_binaryMbo$Kkc > median(NL_binaryMbo$Kkc)] <- 1
NL_binaryMbo$Kkp[NL_binaryMbo$Kkp <= median(NL_binaryMbo$Kkp)] <- 0
NL_binaryMbo$Kkp[NL_binaryMbo$Kkp > median(NL_binaryMbo$Kkp)] <- 1
NL_binaryMbo$Ksp[NL_binaryMbo$Ksp <= median(NL_binaryMbo$Ksp)] <- 0
NL_binaryMbo$Ksp[NL_binaryMbo$Ksp > median(NL_binaryMbo$Ksp)] <- 1
NL_binaryMbo$Ksl[NL_binaryMbo$Ksl <= median(NL_binaryMbo$Ksl)] <- 0
NL_binaryMbo$Ksl[NL_binaryMbo$Ksl > median(NL_binaryMbo$Ksl)] <- 1
NL_binaryMbo$Kse[NL_binaryMbo$Kse <= median(NL_binaryMbo$Kse)] <- 0
NL_binaryMbo$Kse[NL_binaryMbo$Kse > median(NL_binaryMbo$Kse)] <- 1

# Save
save(NL_binaryMbo, file = "NL_binaryMbo.Rda")
```

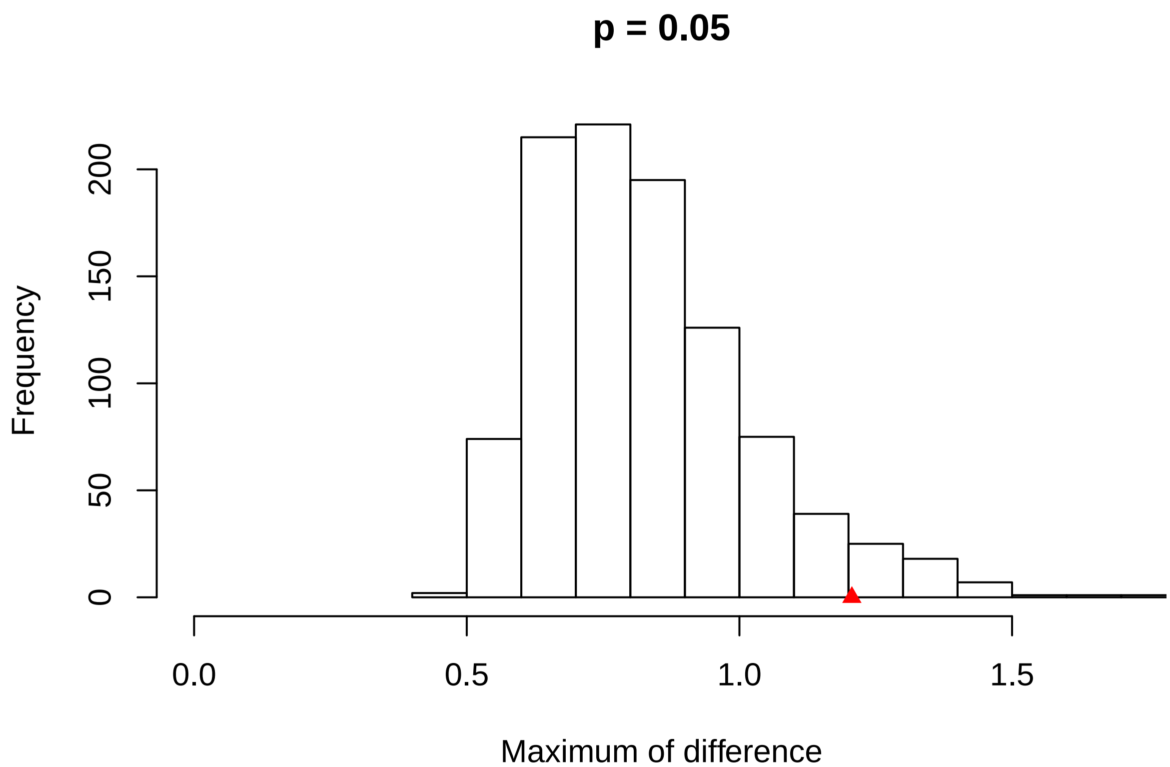
NL Havo/Vwo vs. MBO NCT

```r
load("NL_binaryMbo.Rda")
load("NL_binary.Rda")  # all used Vs

HavoVwoBinary <- na.omit(NL_binary[, -c(1, 2, 42:44)])
MboBinary <- na.omit(NL_binaryMbo[, -c(1, 2, 42:44)])

# NL_NCT_HavoVwoVsMbo <- NCT(HavoVwoBinary, MboBinary, it = 1000, gamma =
# .25, binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges = 'all')
# save(NL_NCT_HavoVwoVsMbo, file='NL_NCT_HavoVwoVsMbo.Rda')
```

```r
load("NL_NCT_HavoVwoVsMbo.Rda")
```

NL HAVO/VWO vs. MBO NCT -> invariance

```r
plot(NL_NCT_HavoVwoVsMbo, what = "network")
```




```r
NL_NCT_HavoVwoVsMbo$nwinv.real
```

```
## [1] 1.2062
```

```r
NL_NCT_HavoVwoVsMbo$nwinv.pval
```

```
## [1] 0.05
```

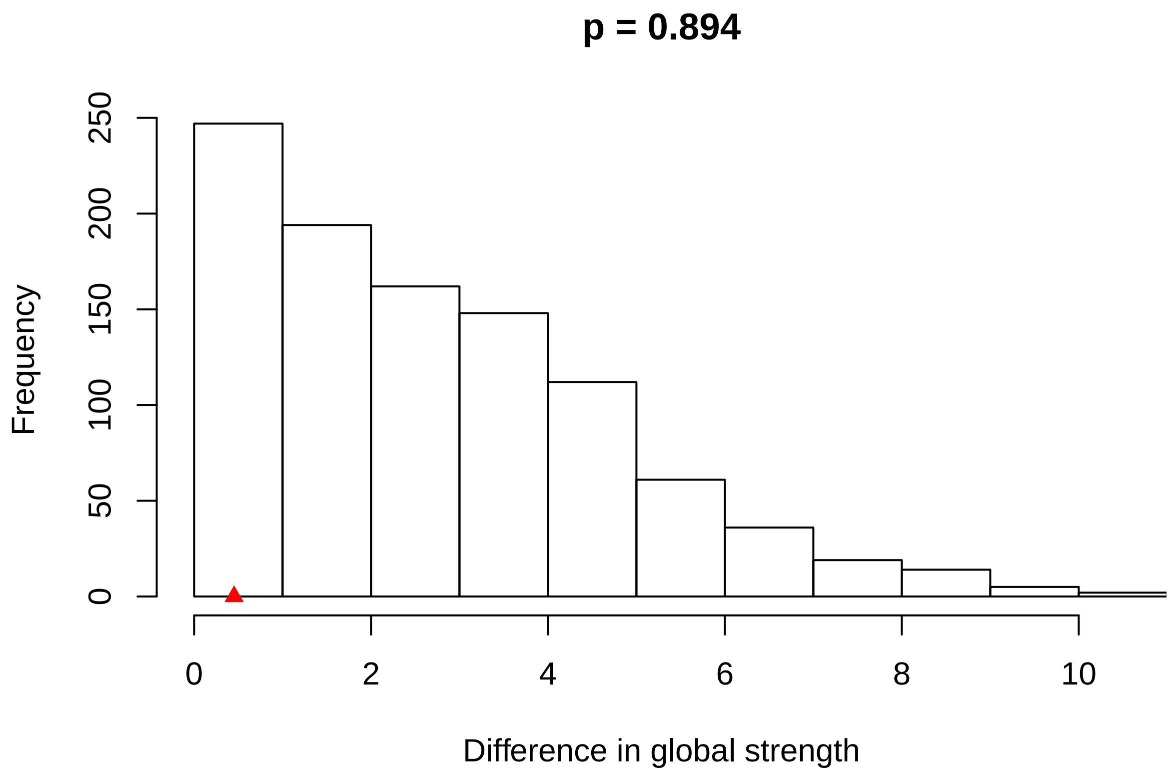
NL HAVO/VWO vs. MBO NCT -> specific edges

```r
p.edges <- NL_NCT_HavoVwoVsMbo$einv.pvals
sign.edges <- p.edges[p.edges$`p-value` < 0.05, ]
sign.edges  # shows which edges are significant
```

```
##      Var1 Var2 p-value
## 240   Ibi  Imf       0
## 440   Vwl  Vdo       0
## 478   Ipd  Vcp       0
## 757   Bbo  Bsn       0
## 960   Sne  Sea       0
## 1200  Slm  Sfr       0
## 1234  Sea  Kce       0
## 1448  Eil  Ksl       0
```

NL HAVO/VWO vs. MBO NCT -> global strength

```r
# global strength
plot(NL_NCT_HavoVwoVsMbo, what = "strength")
```




```r
NL_NCT_HavoVwoVsMbo$glstrinv.real
```

```
## [1] 0.4521982
```

```r
NL_NCT_HavoVwoVsMbo$glstrinv.pval
```

```
## [1] 0.894
```

```r
# global strength per network
NL_NCT_HavoVwoVsMbo$glstrinv.sep
```

```
## [1] 98.01769 98.46989
```

Col

Col data preparation

```r
#Col_complete <- dataset[dataset$CNT=="Colombia",]
#save(Col_complete, file = "Col_complete.Rda")

# getting data ready
load("Col_complete.Rda")

# Recode variables - creating data set per interest-related construct
# Student ID: CNTSTUID, Gender: ST004D01T
```

```r
dataDemo <- data.frame(ID = Col_complete$CNTSTUID,
                       Gen = as.numeric(as.character(revalue(Col_complete$ST004D01T,
                                               c("Male" = "0", "Female" = "1")))))
# enjoyment items:ST094Q01NA-ST094Q05NA
dataEnj <- data.frame(Efu = as.numeric(as.character(revalue(Col_complete$ST094Q01NA,
                                               c("Strongly disagree" = "1",
                                                 "Disagree" = "2",
                                                 "Agree" = "3",
                                                 "Strongly agree" = "4")))),
                      Elr = as.numeric(as.character(revalue(Col_complete$ST094Q02NA,
                                               c("Strongly disagree" = "1",
                                                 "Disagree" = "2",
                                                 "Agree" = "3",
                                                 "Strongly agree" = "4")))),
                      Ehw = as.numeric(as.character(revalue(Col_complete$ST094Q03NA,
                                               c("Strongly disagree" = "1",
                                                 "Disagree" = "2",
                                                 "Agree" = "3",
                                                 "Strongly agree" = "4")))),
                      Eac = as.numeric(as.character(revalue(Col_complete$ST094Q04NA,
                                                 c("Strongly disagree" = "1",
                                                   "Disagree" = "2",
                                                   "Agree" = "3",
                                                   "Strongly agree" = "4")))),
                      Eil = as.numeric(as.character(revalue(Col_complete$ST094Q05NA,
                                                 c("Strongly disagree" = "1",
                                                   "Disagree" = "2",
                                                   "Agree" = "3",
                                                   "Strongly agree" = "4")))))
# interest items: ST095Q04NA-ST095Q15NA
dataInt <- data.frame(Ibi = as.numeric(as.character(revalue(Col_complete$ST095Q04NA,
                                                  c("Not interested" = "1",
                                                    "Hardly interested" = "2",
                                                    "Interested" = "3",
                                                    "Highly interested" = "4",
                                                    "I don't know what this is" = "NA")))),
                      Imf = as.numeric(as.character(revalue(Col_complete$ST095Q07NA,
                                                  c("Not interested" = "1",
                                                    "Hardly interested" = "2",
                                                    "Interested" = "3",
                                                    "Highly interested" = "4",
                                                    "I don't know what this is" = "NA")))),
                      Iet = as.numeric(as.character(revalue(Col_complete$ST095Q08NA,
                                                  c("Not interested" = "1",
                                                    "Hardly interested" = "2",
                                                    "Interested" = "3",
                                                    "Highly interested" = "4",
                                                    "I don't know what this is" = "NA")))),
                      Iun = as.numeric(as.character(revalue(Col_complete$ST095Q13NA,
                                                  c("Not interested" = "1",
                                                    "Hardly interested" = "2",
                                                    "Interested" = "3",
                                                    "Highly interested" = "4",
```

```
                                               "I don't know what this is" = "NA")))),
                      Ipd = as.numeric(as.character(revalue(Col_complete$ST095Q15NA,
                                             c("Not interested" = "1",
                                               "Hardly interested" = "2",
                                               "Interested" = "3",
                                               "Highly interested" = "4",
                                               "I don't know what this is" = "NA")))))
# value items: ST113Q01TA-ST113Q04TA
dataVal <- data.frame(Vwl = as.numeric(as.character(revalue(Col_complete$ST113Q01TA,
                                             c("Strongly Disagree" = "1",
                                               "Disagree" = "2",
                                               "Agree" = "3",
                                               "Strongly Agree" = "4")))),
                      Vdo = as.numeric(as.character(revalue(Col_complete$ST113Q02TA,
                                             c("Strongly Disagree" = "1",
                                              "Disagree" = "2",
                                              "Agree" = "3",
                                              "Strongly Agree" = "4")))),
                      Vcp = as.numeric(as.character(revalue(Col_complete$ST113Q03TA,
                                             c("Strongly Disagree" = "1",
                                               "Disagree" = "2",
                                               "Agree" = "3",
                                               "Strongly Agree" = "4")))),
                      Vhp = as.numeric(as.character(revalue(Col_complete$ST113Q04TA,
                                             c("Strongly Disagree" = "1",
                                               "Disagree" = "2",
                                               "Agree" = "3",
                                              "Strongly Agree" = "4")))))
# behavior items: ST146Q01TA-ST146Q09NA
dataBeh <- data.frame(Btv = as.numeric(as.character(revalue(Col_complete$ST146Q01TA,
                                             c("Never or hardly ever" = "1",
                                               "Sometimes" = "2",
                                               "Regularly" = "3",
                                               "Very often" = "4")))),
                      Bbo = as.numeric(as.character(revalue(Col_complete$ST146Q02TA,
                                             c("Never or hardly ever" = "1",
                                               "Sometimes" = "2",
                                               "Regularly" = "3",
                                               "Very often" = "4")))),
                      Bws = as.numeric(as.character(revalue(Col_complete$ST146Q03TA,
                                             c("Never or hardly ever" = "1",
                                               "Sometimes" = "2",
                                               "Regularly" = "3",
                                               "Very often" = "4")))),
                      Bre = as.numeric(as.character(revalue(Col_complete$ST146Q04TA,
                                             c("Never or hardly ever" = "1",
                                               "Sometimes" = "2",
                                               "Regularly" = "3",
                                               "Very often" = "4")))),
                      Bsc = as.numeric(as.character(revalue(Col_complete$ST146Q05TA,
                                             c("Never or hardly ever" = "1",
                                               "Sometimes" = "2",
                                               "Regularly" = "3",
```

```
                                                            "Very often" = "4")))),
                      Bsn = as.numeric(as.character(revalue(Col_complete$ST146Q06NA,
                                                          c("Never or hardly ever" = "1",
                                                            "Sometimes" = "2",
                                                            "Regularly" = "3",
                                                            "Very often" = "4")))),
                      Bst = as.numeric(as.character(revalue(Col_complete$ST146Q07NA,
                                                          c("Never or hardly ever" = "1",
                                                            "Sometimes" = "2",
                                                            "Regularly" = "3",
                                                            "Very often" = "4")))),
                      Bwe = as.numeric(as.character(revalue(Col_complete$ST146Q08NA,
                                                          c("Never or hardly ever" = "1",
                                                            "Sometimes" = "2",
                                                            "Regularly" = "3",
                                                            "Very often" = "4")))),
                      Bnb = as.numeric(as.character(revalue(Col_complete$ST146Q09NA,
                                                          c("Never or hardly ever" = "1",
                                                            "Sometimes" = "2",
                                                            "Regularly" = "3",
                                                            "Very often" = "4")))))
# science self-efficacy items
dataSel <- data.frame(Sne = as.numeric(as.character(revalue(Col_complete$ST129Q01TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Sea = as.numeric(as.character(revalue(Col_complete$ST129Q02TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Sad = as.numeric(as.character(revalue(Col_complete$ST129Q03TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Sdg = as.numeric(as.character(revalue(Col_complete$ST129Q04TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Scs = as.numeric(as.character(revalue(Col_complete$ST129Q05TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Slf = as.numeric(as.character(revalue(Col_complete$ST129Q06TA,
                                                  c("I could do this easily" = "4",
                                                    "I could do this with a bit of effort" = "3",
                                                    "I would struggle to do this on my own" = "2",
                                                    "I couldn't do this" = "1")))),
                      Slm = as.numeric(as.character(revalue(Col_complete$ST129Q07TA,
```

```
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
                                           "I would struggle to do this on my own" = "2",
                                           "I couldn't do this" = "1")))),
                      Sfr = as.numeric(as.character(revalue(Col_complete$ST129Q08TA,
                                         c("I could do this easily" = "4",
                                           "I could do this with a bit of effort" = "3",
                                           "I would struggle to do this on my own" = "2",
                                           "I couldn't do this" = "1")))))

# knowledge: 1 PV per subscale
dataKno <- data.frame(Kce = as.numeric(Col_complete$PV10SCEP),
                      Kcd = as.numeric(Col_complete$PV10SCED),
                      Kci = as.numeric(Col_complete$PV10SCID),
                      Kkc = as.numeric(Col_complete$PV10SKCO),
                      Kkp = as.numeric(Col_complete$PV10SKPE),
                      Ksp = as.numeric(Col_complete$PV10SSPH),
                      Ksl = as.numeric(Col_complete$PV10SSLI),
                      Kse = as.numeric(Col_complete$PV10SSES))

### SES measure - ESCS
SES <- as.numeric(Col_complete$ESCS)

### school level - PROGN
dataLev <- data.frame(Lev = as.numeric(as.character(revalue(Col_complete$PROGN,
                          c("Colombia: Secondary education  (lower)" = "0",
                            "Colombia: Secondary education (upper), academica" = "1",
                            "Colombia: Secondary education (upper), tecnica" = "2")))))
### Age
dataAge <- data.frame(Age = as.numeric(as.character(Col_complete$AGE)))

# Combining all needed variables in one dataframe:
Col_networkData <- data.frame(dataDemo, dataEnj, dataInt, dataVal, dataBeh, dataSel, dataKno, SES, dataL
save(Col_networkData, file = "Col_networkData.Rda")
```

Creating Col_small by randomly sampling data from Col_big

```
load("Col_networkData.Rda")
# dataset from upper secondary school:
Col_nonBinary_big <- na.omit(Col_networkData[Col_networkData$Lev==1|
                                             Col_networkData$Lev==2,])
save(Col_nonBinary_big, file = "Col_nonBinary_big.Rda")

# smaller sample, so that NL and Col are comparable:
#set.seed(1)
#Col_nonBinary_small <- Col_nonBinary_big[sample(1:5557, 2129, replace = FALSE),]
#save(Col_nonBinary_small, file = "Col_nonBinary_small.Rda")
```

Col_small binarizing

```
load("Col_nonBinary_small.Rda")  # all used Vs and weight
Col_binary_small <- Col_nonBinary_small

### binarize variables
# Enjoyment:
Col_binary_small$Efu[Col_binary_small$Efu<3]  <- 0
Col_binary_small$Efu[Col_binary_small$Efu>2]  <- 1
Col_binary_small$Elr[Col_binary_small$Elr<3]  <- 0
Col_binary_small$Elr[Col_binary_small$Elr>2]  <- 1
Col_binary_small$Ehw[Col_binary_small$Ehw<3]  <- 0
Col_binary_small$Ehw[Col_binary_small$Ehw>2]  <- 1
Col_binary_small$Eac[Col_binary_small$Eac<3]  <- 0
Col_binary_small$Eac[Col_binary_small$Eac>2]  <- 1
Col_binary_small$Eil[Col_binary_small$Eil<3]  <- 0
Col_binary_small$Eil[Col_binary_small$Eil>2]  <- 1

# interest:
Col_binary_small$Ibi[Col_binary_small$Ibi<3]  <- 0
Col_binary_small$Ibi[Col_binary_small$Ibi>2]  <- 1
Col_binary_small$Imf[Col_binary_small$Imf<3]  <- 0
Col_binary_small$Imf[Col_binary_small$Imf>2]  <- 1
Col_binary_small$Iet[Col_binary_small$Iet<3]  <- 0
Col_binary_small$Iet[Col_binary_small$Iet>2]  <- 1
Col_binary_small$Iun[Col_binary_small$Iun<3]  <- 0
Col_binary_small$Iun[Col_binary_small$Iun>2]  <- 1
Col_binary_small$Ipd[Col_binary_small$Ipd<3]  <- 0
Col_binary_small$Ipd[Col_binary_small$Ipd>2]  <- 1

# Value:
Col_binary_small$Vwl[Col_binary_small$Vwl<3]  <- 0
Col_binary_small$Vwl[Col_binary_small$Vwl>2]  <- 1
Col_binary_small$Vdo[Col_binary_small$Vdo<3]  <- 0
Col_binary_small$Vdo[Col_binary_small$Vdo>2]  <- 1
Col_binary_small$Vcp[Col_binary_small$Vcp<3]  <- 0
Col_binary_small$Vcp[Col_binary_small$Vcp>2]  <- 1
Col_binary_small$Vhp[Col_binary_small$Vhp<3]  <- 0
Col_binary_small$Vhp[Col_binary_small$Vhp>2]  <- 1

# Behavior:
Col_binary_small$Btv[Col_binary_small$Btv<2]  <- 0
Col_binary_small$Btv[Col_binary_small$Btv>1]  <- 1
Col_binary_small$Bbo[Col_binary_small$Bbo<2]  <- 0
Col_binary_small$Bbo[Col_binary_small$Bbo>1]  <- 1
Col_binary_small$Bws[Col_binary_small$Bws<2]  <- 0
Col_binary_small$Bws[Col_binary_small$Bws>1]  <- 1
Col_binary_small$Bre[Col_binary_small$Bre<2]  <- 0
Col_binary_small$Bre[Col_binary_small$Bre>1]  <- 1
Col_binary_small$Bsc[Col_binary_small$Bsc<2]  <- 0
Col_binary_small$Bsc[Col_binary_small$Bsc>1]  <- 1
Col_binary_small$Bsn[Col_binary_small$Bsn<2]  <- 0
Col_binary_small$Bsn[Col_binary_small$Bsn>1]  <- 1
```

```
Col_binary_small$Bst[Col_binary_small$Bst<2]  <- 0
Col_binary_small$Bst[Col_binary_small$Bst>1]  <- 1
Col_binary_small$Bwe[Col_binary_small$Bwe<2]  <- 0
Col_binary_small$Bwe[Col_binary_small$Bwe>1]  <- 1
Col_binary_small$Bnb[Col_binary_small$Bnb<2]  <- 0
Col_binary_small$Bnb[Col_binary_small$Bnb>1]  <- 1

# Self efficacy
Col_binary_small$Sne[Col_binary_small$Sne<3]  <- 0
Col_binary_small$Sne[Col_binary_small$Sne>2]  <- 1
Col_binary_small$Sea[Col_binary_small$Sea<3]  <- 0
Col_binary_small$Sea[Col_binary_small$Sea>2]  <- 1
Col_binary_small$Sad[Col_binary_small$Sad<3]  <- 0
Col_binary_small$Sad[Col_binary_small$Sad>2]  <- 1
Col_binary_small$Sdg[Col_binary_small$Sdg<3]  <- 0
Col_binary_small$Sdg[Col_binary_small$Sdg>2]  <- 1
Col_binary_small$Scs[Col_binary_small$Scs<3]  <- 0
Col_binary_small$Scs[Col_binary_small$Scs>2]  <- 1
Col_binary_small$Slf[Col_binary_small$Slf<3]  <- 0
Col_binary_small$Slf[Col_binary_small$Slf>2]  <- 1
Col_binary_small$Slm[Col_binary_small$Slm<3]  <- 0
Col_binary_small$Slm[Col_binary_small$Slm>2]  <- 1
Col_binary_small$Sfr[Col_binary_small$Sfr<3]  <- 0
Col_binary_small$Sfr[Col_binary_small$Sfr>2]  <- 1

# Knowledge -> Median
Col_binary_small$Kce[Col_binary_small$Kce<=median(Col_binary_small$Kce)] <- 0
Col_binary_small$Kce[Col_binary_small$Kce>median(Col_binary_small$Kce)] <- 1
Col_binary_small$Kcd[Col_binary_small$Kcd<=median(Col_binary_small$Kcd)] <- 0
Col_binary_small$Kcd[Col_binary_small$Kcd>median(Col_binary_small$Kcd)] <- 1
Col_binary_small$Kci[Col_binary_small$Kci<=median(Col_binary_small$Kci)] <- 0
Col_binary_small$Kci[Col_binary_small$Kci>median(Col_binary_small$Kci)] <- 1
Col_binary_small$Kkc[Col_binary_small$Kkc<=median(Col_binary_small$Kkc)] <- 0
Col_binary_small$Kkc[Col_binary_small$Kkc>median(Col_binary_small$Kkc)] <- 1
Col_binary_small$Kkp[Col_binary_small$Kkp<=median(Col_binary_small$Kkp)] <- 0
Col_binary_small$Kkp[Col_binary_small$Kkp>median(Col_binary_small$Kkp)] <- 1
Col_binary_small$Ksp[Col_binary_small$Ksp<=median(Col_binary_small$Ksp)] <- 0
Col_binary_small$Ksp[Col_binary_small$Ksp>median(Col_binary_small$Ksp)] <- 1
Col_binary_small$Ksl[Col_binary_small$Ksl<=median(Col_binary_small$Ksl)] <- 0
Col_binary_small$Ksl[Col_binary_small$Ksl>median(Col_binary_small$Ksl)] <- 1
Col_binary_small$Kse[Col_binary_small$Kse<=median(Col_binary_small$Kse)] <- 0
Col_binary_small$Kse[Col_binary_small$Kse>median(Col_binary_small$Kse)] <- 1

# Save
save(Col_binary_small, file="Col_binary_small.Rda")
```

Col_big binarizing

```
load("Col_nonBinary_big.Rda")  # all used Vs and weight
Col_binary_big <- Col_nonBinary_big

### binarize variables
```

```r
# Enjoyment:
Col_binary_big$Efu[Col_binary_big$Efu<3]  <- 0
Col_binary_big$Efu[Col_binary_big$Efu>2]  <- 1
Col_binary_big$Elr[Col_binary_big$Elr<3]  <- 0
Col_binary_big$Elr[Col_binary_big$Elr>2]  <- 1
Col_binary_big$Ehw[Col_binary_big$Ehw<3]  <- 0
Col_binary_big$Ehw[Col_binary_big$Ehw>2]  <- 1
Col_binary_big$Eac[Col_binary_big$Eac<3]  <- 0
Col_binary_big$Eac[Col_binary_big$Eac>2]  <- 1
Col_binary_big$Eil[Col_binary_big$Eil<3]  <- 0
Col_binary_big$Eil[Col_binary_big$Eil>2]  <- 1

# interest:
Col_binary_big$Ibi[Col_binary_big$Ibi<3]  <- 0
Col_binary_big$Ibi[Col_binary_big$Ibi>2]  <- 1
Col_binary_big$Imf[Col_binary_big$Imf<3]  <- 0
Col_binary_big$Imf[Col_binary_big$Imf>2]  <- 1
Col_binary_big$Iet[Col_binary_big$Iet<3]  <- 0
Col_binary_big$Iet[Col_binary_big$Iet>2]  <- 1
Col_binary_big$Iun[Col_binary_big$Iun<3]  <- 0
Col_binary_big$Iun[Col_binary_big$Iun>2]  <- 1
Col_binary_big$Ipd[Col_binary_big$Ipd<3]  <- 0
Col_binary_big$Ipd[Col_binary_big$Ipd>2]  <- 1

# Value:
Col_binary_big$Vwl[Col_binary_big$Vwl<3]  <- 0
Col_binary_big$Vwl[Col_binary_big$Vwl>2]  <- 1
Col_binary_big$Vdo[Col_binary_big$Vdo<3]  <- 0
Col_binary_big$Vdo[Col_binary_big$Vdo>2]  <- 1
Col_binary_big$Vcp[Col_binary_big$Vcp<3]  <- 0
Col_binary_big$Vcp[Col_binary_big$Vcp>2]  <- 1
Col_binary_big$Vhp[Col_binary_big$Vhp<3]  <- 0
Col_binary_big$Vhp[Col_binary_big$Vhp>2]  <- 1

# Behavior:
Col_binary_big$Btv[Col_binary_big$Btv<2]  <- 0
Col_binary_big$Btv[Col_binary_big$Btv>1]  <- 1
Col_binary_big$Bbo[Col_binary_big$Bbo<2]  <- 0
Col_binary_big$Bbo[Col_binary_big$Bbo>1]  <- 1
Col_binary_big$Bws[Col_binary_big$Bws<2]  <- 0
Col_binary_big$Bws[Col_binary_big$Bws>1]  <- 1
Col_binary_big$Bre[Col_binary_big$Bre<2]  <- 0
Col_binary_big$Bre[Col_binary_big$Bre>1]  <- 1
Col_binary_big$Bsc[Col_binary_big$Bsc<2]  <- 0
Col_binary_big$Bsc[Col_binary_big$Bsc>1]  <- 1
Col_binary_big$Bsn[Col_binary_big$Bsn<2]  <- 0
Col_binary_big$Bsn[Col_binary_big$Bsn>1]  <- 1
Col_binary_big$Bst[Col_binary_big$Bst<2]  <- 0
Col_binary_big$Bst[Col_binary_big$Bst>1]  <- 1
Col_binary_big$Bwe[Col_binary_big$Bwe<2]  <- 0
Col_binary_big$Bwe[Col_binary_big$Bwe>1]  <- 1
Col_binary_big$Bnb[Col_binary_big$Bnb<2]  <- 0
Col_binary_big$Bnb[Col_binary_big$Bnb>1]  <- 1
```

```
# Self efficacy
Col_binary_big$Sne[Col_binary_big$Sne<3]  <- 0
Col_binary_big$Sne[Col_binary_big$Sne>2]  <- 1
Col_binary_big$Sea[Col_binary_big$Sea<3]  <- 0
Col_binary_big$Sea[Col_binary_big$Sea>2]  <- 1
Col_binary_big$Sad[Col_binary_big$Sad<3]  <- 0
Col_binary_big$Sad[Col_binary_big$Sad>2]  <- 1
Col_binary_big$Sdg[Col_binary_big$Sdg<3]  <- 0
Col_binary_big$Sdg[Col_binary_big$Sdg>2]  <- 1
Col_binary_big$Scs[Col_binary_big$Scs<3]  <- 0
Col_binary_big$Scs[Col_binary_big$Scs>2]  <- 1
Col_binary_big$Slf[Col_binary_big$Slf<3]  <- 0
Col_binary_big$Slf[Col_binary_big$Slf>2]  <- 1
Col_binary_big$Slm[Col_binary_big$Slm<3]  <- 0
Col_binary_big$Slm[Col_binary_big$Slm>2]  <- 1
Col_binary_big$Sfr[Col_binary_big$Sfr<3]  <- 0
Col_binary_big$Sfr[Col_binary_big$Sfr>2]  <- 1

# Knowledge -> Median
Col_binary_big$Kce[Col_binary_big$Kce<=median(Col_binary_big$Kce)] <- 0
Col_binary_big$Kce[Col_binary_big$Kce>median(Col_binary_big$Kce)] <- 1
Col_binary_big$Kcd[Col_binary_big$Kcd<=median(Col_binary_big$Kcd)] <- 0
Col_binary_big$Kcd[Col_binary_big$Kcd>median(Col_binary_big$Kcd)] <- 1
Col_binary_big$Kci[Col_binary_big$Kci<=median(Col_binary_big$Kci)] <- 0
Col_binary_big$Kci[Col_binary_big$Kci>median(Col_binary_big$Kci)] <- 1
Col_binary_big$Kkc[Col_binary_big$Kkc<=median(Col_binary_big$Kkc)] <- 0
Col_binary_big$Kkc[Col_binary_big$Kkc>median(Col_binary_big$Kkc)] <- 1
Col_binary_big$Kkp[Col_binary_big$Kkp<=median(Col_binary_big$Kkp)] <- 0
Col_binary_big$Kkp[Col_binary_big$Kkp>median(Col_binary_big$Kkp)] <- 1
Col_binary_big$Ksp[Col_binary_big$Ksp<=median(Col_binary_big$Ksp)] <- 0
Col_binary_big$Ksp[Col_binary_big$Ksp>median(Col_binary_big$Ksp)] <- 1
Col_binary_big$Ksl[Col_binary_big$Ksl<=median(Col_binary_big$Ksl)] <- 0
Col_binary_big$Ksl[Col_binary_big$Ksl>median(Col_binary_big$Ksl)] <- 1
Col_binary_big$Kse[Col_binary_big$Kse<=median(Col_binary_big$Kse)] <- 0
Col_binary_big$Kse[Col_binary_big$Kse>median(Col_binary_big$Kse)] <- 1
# Save
save(Col_binary_big, file="Col_binary_big.Rda")
```

Col_big data inspection -> TABLE 2

```
load("Col_nonBinary_big.Rda")  # non-binarized variables

data <- na.omit(Col_nonBinary_big)

boys <- data[data$Gen==0,]
girls <- data[data$Gen==1,]
quantile(data$SES)
```

```
##       0%     25%     50%     75%    100%
## -4.2475 -1.3743 -0.6563  0.1266  2.0187
```

```r
# low SES: within lowest 25%, high SES: within highest 25%:
lowSES <- data[data$SES<=-1.3743,]
highSES <- data[data$SES>=0.1266,]

# Age
mean(data$Age)
```

```
## [1] 15.87918
```

```r
sd(data$Age)
```

```
## [1] 0.2789951
```

```r
mean(boys$Age)
```

```
## [1] 15.88338
```

```r
sd(boys$Age)
```

```
## [1] 0.2768578
```

```r
mean(girls$Age)
```

```
## [1] 15.87597
```

```r
sd(girls$Age)
```

```
## [1] 0.2806177
```

```r
mean(lowSES$Age)
```

```
## [1] 15.89306
```

```r
sd(lowSES$Age)
```

```
## [1] 0.2788631
```

```r
mean(highSES$Age)
```

```
## [1] 15.87235
```

```r
sd(highSES$Age)
```

```
## [1] 0.2742747
```

```r
# knowledge
mean(apply(data[,c(34:41)],2,mean))
```

```
## [1] 454.9874
```

```r
mean(apply(data[,c(34:41)],2,sd))
```

```
## [1] 76.24432
```

```r
mean(apply(boys[,c(34:41)],2,mean))
```

```
## [1] 469.1421
```

```r
mean(apply(boys[,c(34:41)],2,sd))
```

```
## [1] 76.38676
```

```r
mean(apply(girls[,c(34:41)],2,mean))
```

```
## [1] 444.1793
```

```r
mean(apply(girls[,c(34:41)],2,sd))
```

```
## [1] 74.30468
```

```r
mean(apply(lowSES[,c(34:41)],2,mean))
```

```
## [1] 419.71
```

```r
mean(apply(lowSES[,c(34:41)],2,sd))
```

```
## [1] 67.03869
```

```r
mean(apply(highSES[,c(34:41)],2,mean))
```

```
## [1] 493.3983
```

```r
mean(apply(highSES[,c(34:41)],2,sd))
```

```
## [1] 72.82063
```

```r
# Interest
mean(apply(data[,c(8:12)],2,mean))
```

```
## [1] 2.86867
```

```r
mean(apply(data[,c(8:12)],2,sd))
```

```
## [1] 0.8768918
```

```r
mean(apply(boys[,c(8:12)],2,mean))
```

```
## [1] 2.898337
```

```r
mean(apply(boys[,c(8:12)],2,sd))
```

```
## [1] 0.8785606
```

```r
mean(apply(girls[,c(8:12)],2,mean))
```

```
## [1] 2.846017
```

```r
mean(apply(girls[,c(8:12)],2,sd))
```

```
## [1] 0.8704419
```

```r
mean(apply(lowSES[,c(8:12)],2,mean))
```

```
## [1] 2.828489
```

```r
mean(apply(lowSES[,c(8:12)],2,sd))
```

```
## [1] 0.8733192
```

```r
mean(apply(highSES[,c(8:12)],2,mean))
```

```
## [1] 2.902878
```

```r
mean(apply(highSES[,c(8:12)],2,sd))
```

```
## [1] 0.894758
```

```r
# Value
mean(apply(data[,c(13:16)],2,mean))
```

```
## [1] 2.933192
```

```r
mean(apply(data[,c(13:16)],2,sd))
```

```
## [1] 0.7974646
```

```r
mean(apply(boys[,c(13:16)],2,mean))
```

```
## [1] 2.917914
```

```r
mean(apply(boys[,c(13:16)],2,sd))
```

```
## [1] 0.8013879
```

```r
mean(apply(girls[,c(13:16)],2,mean))
```

```
## [1] 2.944859
```

```r
mean(apply(girls[,c(13:16)],2,sd))
```

```
## [1] 0.7940749
```

```r
mean(apply(lowSES[,c(13:16)],2,mean))
```

```
## [1] 3.007374
```

```r
mean(apply(lowSES[,c(13:16)],2,sd))
```

```
## [1] 0.7543998
```

```r
mean(apply(highSES[,c(13:16)],2,mean))
```

```
## [1] 2.901079
```

```r
mean(apply(highSES[,c(13:16)],2,sd))
```

```
## [1] 0.8249405
```

```r
# Enjoyment
mean(apply(data[,c(3:7)],2,mean))
```

```
## [1] 2.818715
```

```r
mean(apply(data[,c(3:7)],2,sd))
```

```
## [1] 0.7672736
```

```r
mean(apply(boys[,c(3:7)],2,mean))
```

```
## [1] 2.810474
```

```r
mean(apply(boys[,c(3:7)],2,sd))
```

```
## [1] 0.7890678
```

```r
mean(apply(girls[,c(3:7)],2,mean))
```

```
## [1] 2.825008
```

```r
mean(apply(girls[,c(3:7)],2,sd))
```

```
## [1] 0.7501019
```

```r
mean(apply(lowSES[,c(3:7)],2,mean))
```

```
## [1] 2.798993
```

```r
mean(apply(lowSES[,c(3:7)],2,sd))
```

```
## [1] 0.7585818
```

```r
mean(apply(highSES[,c(3:7)],2,mean))
```

```
## [1] 2.834245
```

```r
mean(apply(highSES[,c(3:7)],2,sd))
```

```
## [1] 0.7973932
```

```r
# Behavior
mean(apply(data[,c(17:25)],2,mean))
```

```
## [1] 1.828904
```

```r
mean(apply(data[,c(17:25)],2,sd))
```

```
## [1] 0.8842021
```

```r
mean(apply(boys[,c(17:25)],2,mean))
```

```
## [1] 1.914427
```

```r
mean(apply(boys[,c(17:25)],2,sd))
```

```
## [1] 0.9142814
```

```r
mean(apply(girls[,c(17:25)],2,mean))
```

```
## [1] 1.763602
```

```r
mean(apply(girls[,c(17:25)],2,sd))
```

```
## [1] 0.8535676
```

```r
mean(apply(lowSES[,c(17:25)],2,mean))
```

```
## [1] 1.81311
```

```r
mean(apply(lowSES[,c(17:25)],2,sd))
```

```
## [1] 0.8798091
```

```r
mean(apply(highSES[,c(17:25)],2,mean))
```

```
## [1] 1.845803
```

```r
mean(apply(highSES[,c(17:25)],2,sd))
```

```
## [1] 0.88294
```

```r
# Self-Efficacy
mean(apply(data[,c(26:33)],2,mean))
```

```
## [1] 2.662363
```

```r
mean(apply(data[,c(26:33)],2,sd))
```

```
## [1] 0.8755177
```

```r
mean(apply(boys[,c(26:33)],2,mean))
```

```
## [1] 2.669732
```

```r
mean(apply(boys[,c(26:33)],2,sd))
```

```
## [1] 0.8522875
```

```r
mean(apply(girls[,c(26:33)],2,mean))
```

```
## [1] 2.656736
```

```r
mean(apply(girls[,c(26:33)],2,sd))
```

```
## [1] 0.8921985
```

```r
mean(apply(lowSES[,c(26:33)],2,mean))
```

```
## [1] 2.61196
```

```r
mean(apply(lowSES[,c(26:33)],2,sd))
```

```
## [1] 0.8818022
```

```r
mean(apply(highSES[,c(26:33)],2,mean))
```

```
## [1] 2.745504
```

```r
mean(apply(highSES[,c(26:33)],2,sd))
```

```
## [1] 0.8639549
```

```r
# for t-tests
mKnoA <- apply(data[,c(34:41)],1,mean)  # all
sKnoA <- apply(data[,c(34:41)],1,sd)
mKnoB <- apply(boys[,c(34:41)],1,mean) # boys
sKnoB <- apply(boys[,c(34:41)],1,sd)
mKnoG <- apply(girls[,c(34:41)],1,mean) # girls
sKnoG <- apply(girls[,c(34:41)],1,sd)
mKnoL <- apply(lowSES[,c(34:41)],1,mean) # low SES
sKnoL <- apply(lowSES[,c(34:41)],1,sd)
mKnoH <- apply(highSES[,c(34:41)],1,mean) # high SES
sKnoH <- apply(highSES[,c(34:41)],1,sd)

# Interest
mIntA <- apply(data[,c(8:12)],1,mean)
sIntA <- apply(data[,c(8:12)],1,sd)
mIntB <- apply(boys[,c(8:12)],1,mean)
sIntB <- apply(boys[,c(8:12)],1,sd)
mIntG <- apply(girls[,c(8:12)],1,mean)
sIntG <- apply(girls[,c(8:12)],1,sd)
mIntL <- apply(lowSES[,c(8:12)],1,mean)
sIntL <- apply(lowSES[,c(8:12)],1,sd)
mIntH <- apply(highSES[,c(8:12)],1,mean)
sIntH <- apply(highSES[,c(8:12)],1,sd)

# Value
mValA <- apply(data[,c(13:16)],1,mean)
sValA <- apply(data[,c(13:16)],1,sd)
mValB <- apply(boys[,c(13:16)],1,mean)
sValB <- apply(boys[,c(13:16)],1,sd)
mValG <- apply(girls[,c(13:16)],1,mean)
sValG <- apply(girls[,c(13:16)],1,sd)
```

```r
mValL <- apply(lowSES[,c(13:16)],1,mean)
sValL <- apply(lowSES[,c(13:16)],1,sd)
mValH <- apply(highSES[,c(13:16)],1,mean)
sValH <- apply(highSES[,c(13:16)],1,sd)

# Enjoyment
mEnjA <- apply(data[,c(3:7)],1,mean)
sEnjA <- apply(data[,c(3:7)],1,sd)
mEnjB <- apply(boys[,c(3:7)],1,mean)
sEnjB <- apply(boys[,c(3:7)],1,sd)
mEnjG <- apply(girls[,c(3:7)],1,mean)
sEnjG <- apply(girls[,c(3:7)],1,sd)
mEnjL <- apply(lowSES[,c(3:7)],1,mean)
sEnjL <- apply(lowSES[,c(3:7)],1,sd)
mEnjH <- apply(highSES[,c(3:7)],1,mean)
sEnjH <- apply(highSES[,c(3:7)],1,sd)

# Behavior
mBehA <- apply(data[,c(17:25)],1,mean)
sBehA <- apply(data[,c(17:25)],1,sd)
mBehB <- apply(boys[,c(17:25)],1,mean)
sBehB <- apply(boys[,c(17:25)],1,sd)
mBehG <- apply(girls[,c(17:25)],1,mean)
sBehG <- apply(girls[,c(17:25)],1,sd)
mBehL <- apply(lowSES[,c(17:25)],1,mean)
sBehL <- apply(lowSES[,c(17:25)],1,sd)
mBehH <- apply(highSES[,c(17:25)],1,mean)
sBehH <- apply(highSES[,c(17:25)],1,sd)

# Self-Efficacy
mSelA <- apply(data[,c(26:33)],1,mean)
sSelA <- apply(data[,c(26:33)],1,sd)
mSelB <- apply(boys[,c(26:33)],1,mean)
sSelB <- apply(boys[,c(26:33)],1,sd)
mSelG <- apply(girls[,c(26:33)],1,mean)
sSelG <- apply(girls[,c(26:33)],1,sd)
mSelL <- apply(lowSES[,c(26:33)],1,mean)
sSelL <- apply(lowSES[,c(26:33)],1,sd)
mSelH <- apply(highSES[,c(26:33)],1,mean)
sSelH <- apply(highSES[,c(26:33)],1,sd)

# knowledge
t.test(mKnoB, mKnoG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mKnoB and mKnoG
## t = 13.139, df = 5076.8, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  21.23825 28.68752
## sample estimates:
## mean of x mean of y
```

```
## 469.1421 444.1793
```

```r
t.test(mKnoL, mKnoH, var.equal = FALSE) # perform t-test on the SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mKnoL and mKnoH
## t = -30.256, df = 2750.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -78.46393 -68.91268
## sample estimates:
## mean of x mean of y
##  419.7100  493.3983
```

```r
# Interest
t.test(mIntB, mIntG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mIntB and mIntG
## t = 2.994, df = 5143.9, p-value = 0.002766
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01806222 0.08657848
## sample estimates:
## mean of x mean of y
##  2.898337  2.846017
```

```r
t.test(mIntL, mIntH, var.equal = FALSE) # perform t-test on SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mIntL and mIntH
## t = -2.9927, df = 2769.7, p-value = 0.00279
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.12312844 -0.02564854
## sample estimates:
## mean of x mean of y
##  2.828489  2.902878
```

```r
# Value
t.test(mValB, mValG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mValB and mValG
## t = -1.4248, df = 5188.1, p-value = 0.1543
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06401984  0.01012938
## sample estimates:
```

```
## mean of x mean of y
##  2.917914  2.944859
```

```r
t.test(mValL, mValH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mValL and mValH
## t = 4.0342, df = 2750.3, p-value = 5.628e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.05463007 0.15795986
## sample estimates:
## mean of x mean of y
##  3.007374  2.901079
```

```r
# Enjoyment
t.test(mEnjB, mEnjG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mEnjB and mEnjG
## t = -0.80448, df = 5023.8, p-value = 0.4212
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.04995231  0.02088407
## sample estimates:
## mean of x mean of y
##  2.810474  2.825008
```

```r
t.test(mEnjL, mEnjH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mEnjL and mEnjH
## t = -1.3754, df = 2754.4, p-value = 0.1691
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08550719  0.01500359
## sample estimates:
## mean of x mean of y
##  2.798993  2.834245
```

```r
# Behavior
t.test(mBehB, mBehG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mBehB and mBehG
## t = 8.3458, df = 4926.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.1153955 0.1862535
```

```
## sample estimates:
## mean of x mean of y
##  1.914427  1.763602
```

```r
t.test(mBehL, mBehH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mBehL and mBehH
## t = -1.2883, df = 2772, p-value = 0.1977
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.08245336  0.01706567
## sample estimates:
## mean of x mean of y
##  1.813110  1.845803
```

```r
# Self-Efficacy
t.test(mSelB, mSelG, var.equal = FALSE) # perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mSelB and mSelG
## t = 0.7534, df = 5289.4, p-value = 0.4512
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02082056  0.04681249
## sample estimates:
## mean of x mean of y
##  2.669732  2.656736
```

```r
t.test(mSelL, mSelH, var.equal = FALSE) # perform t-test SES difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mSelL and mSelH
## t = -5.486, df = 2765.6, p-value = 4.484e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1812746 -0.0858117
## sample estimates:
## mean of x mean of y
##  2.611960  2.745504
```

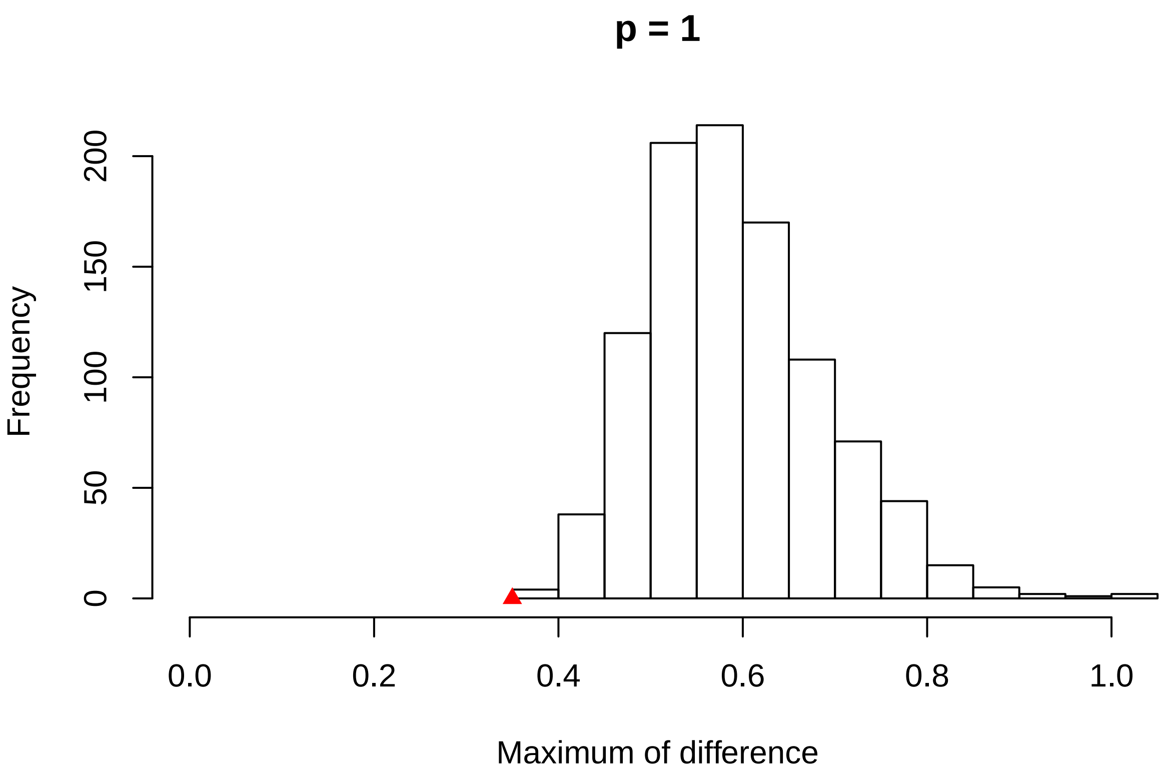
NCT Col_small vs Col_big

```r
load("Col_binary_big.Rda")
load("Col_binary_small.Rda")

Col_data_small <- Col_binary_small[, -c(1, 2, 42:44)]
Col_data_big <- Col_binary_big[, -c(1, 2, 42:44)]
```

```r
# Network comparison test Col_NCT_Size <- NCT(Col_data_small, Col_data_big,
# it = 1000, gamma = .25, binary.data = TRUE, paired = FALSE, test.edges =
# TRUE, edges = 'all') save(Col_NCT_Size, file='Col_NCT_Size.Rda')
load("Col_NCT_Size.Rda")
```

Col small vs Col big NCT -> invariance

```r
plot(Col_NCT_Size, what = "network")
```




```r
Col_NCT_Size$nwinv.real
```

```
## [1] 0.3499668
```

```r
Col_NCT_Size$nwinv.pval
```

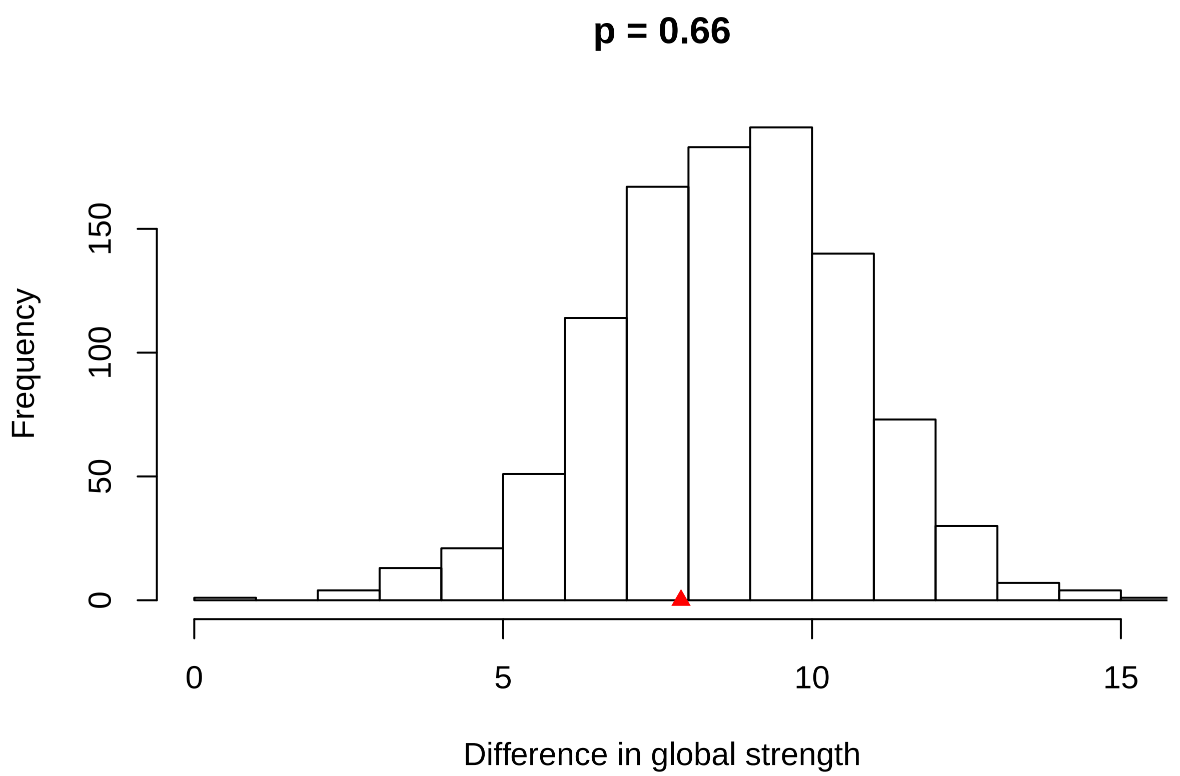
```
## [1] 1
```

Col small vs Col big NCT -> specific edges; not invariant -> do not test specific edges

```r
#p.edges <- Col_NCT_Size$einv.pvals
#which(p.edges$`p-value`<0.05) #
```

Col small vs Col big NCT -> global strength

```r
# global strength
plot(Col_NCT_Size, what = "strength")
```




```
Col_NCT_Size$glstrinv.real
```

```
## [1] 7.879629
```

```
Col_NCT_Size$glstrinv.pval
```

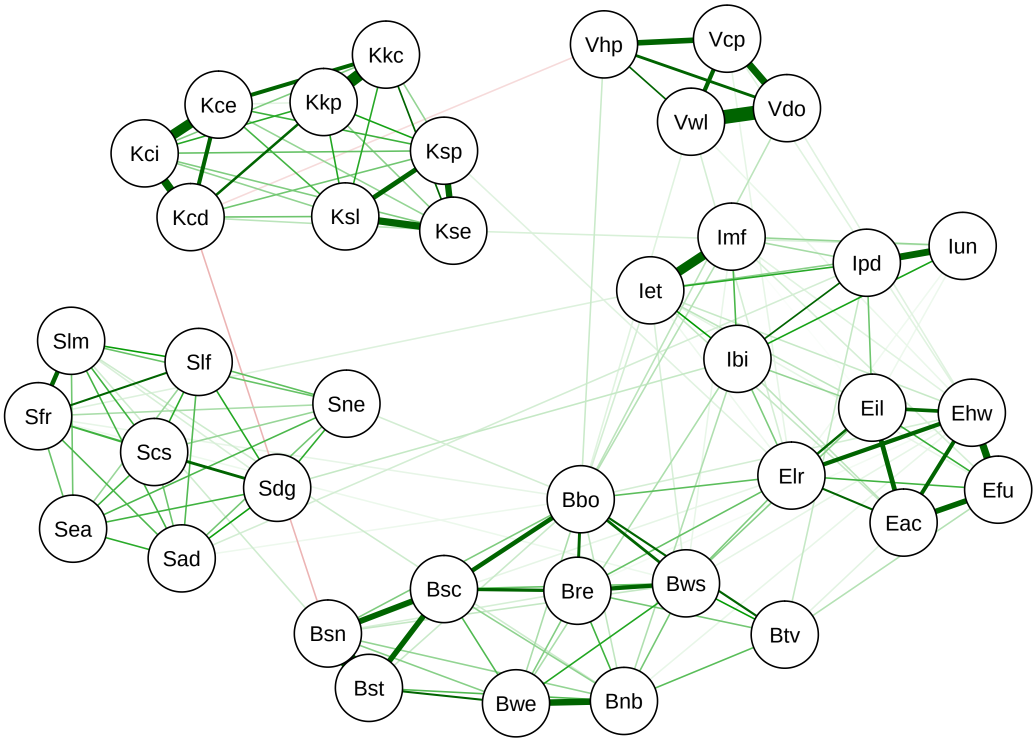
```
## [1] 0.66
```

```
# global strength per network
Col_NCT_Size$glstrinv.sep
```

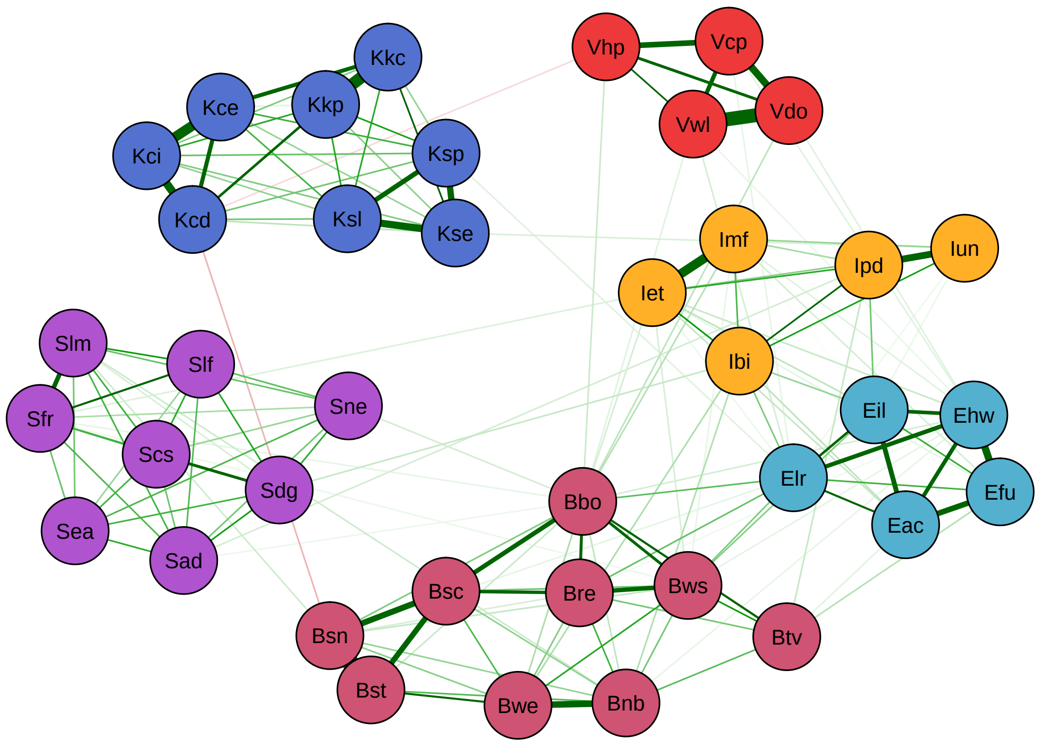
```
## [1] 88.64384 96.52347
```

Col_big Network estimation

```
load("Col_binary_big.Rda")
Col_binary_big <- na.omit(Col_binary_big[,-c(1,2,42:44)])

# Network estimation
Col_fit <- IsingFit(Col_binary_big)
```




```
Col_graph <- qgraph(Col_fit$weiadj,
                    layout = 'spring',
                    color = group_col,
                    groups = groups_type,
                    nodeNames = names,
                    legend = FALSE,
                    legend.mode="style2",
                    legend.cex=.3)
```



Col_big Community detection - for graph see NL vs COL

```
Col_igraph <- graph_from_adjacency_matrix(abs(Col_fit$weiadj),
                                          'undirected',
                                          weighted = TRUE,
                                          add.colnames = FALSE)
Col_Com <- cluster_walktrap(Col_igraph)
communities(Col_Com)
```

```
## $`1`
## [1] 15 16 17 18 19 20 21 22 23
##
## $`2`
## [1] 1 2 3 4 5
##
## $`3`
## [1]  6  7  8  9 10
##
## $`4`
## [1] 32 33 34 35 36 37 38 39
##
## $`5`
## [1] 24 25 26 27 28 29 30 31
##
## $`6`
## [1] 11 12 13 14
```

Col_big Centrality -> see Nl vs Col

Col_big Smallworldness

Col_big Small worldness -> as a double check

```
smallworldness(Col_graph, B = 1000, up = .995, lo = .005)
```

```
##       smallworldness         trans_target averagelength_target
##            2.1685279            0.5862832            2.2550607
##          trans_rnd_M         trans_rnd_lo         trans_rnd_up
##            0.2262544            0.1902655            0.2654867
##  averagelength_rnd_M averagelength_rnd_lo averagelength_rnd_up
##            1.8871781            1.8596491            1.9203846
```

Col_big smallworldness: get upper CI

```
SW_Index(Col_igraph)
```

```
## $SW.Index
## [1] 2.245307
##
## $Upper.CI
##      CI Value.CI
## 1 0.100 1.126302
## 2 0.050 1.150323
## 3 0.010 1.196404
## 4 0.001 1.249196
##
## $Clustering.Graph
## [1] 0.5862832
##
## $Clustering.Random.Graph
## [1] 0.219702
##
## $ASPL.Graph
## [1] 2.255061
##
## $ASPL.Random.Graph
## [1] 1.897407
```

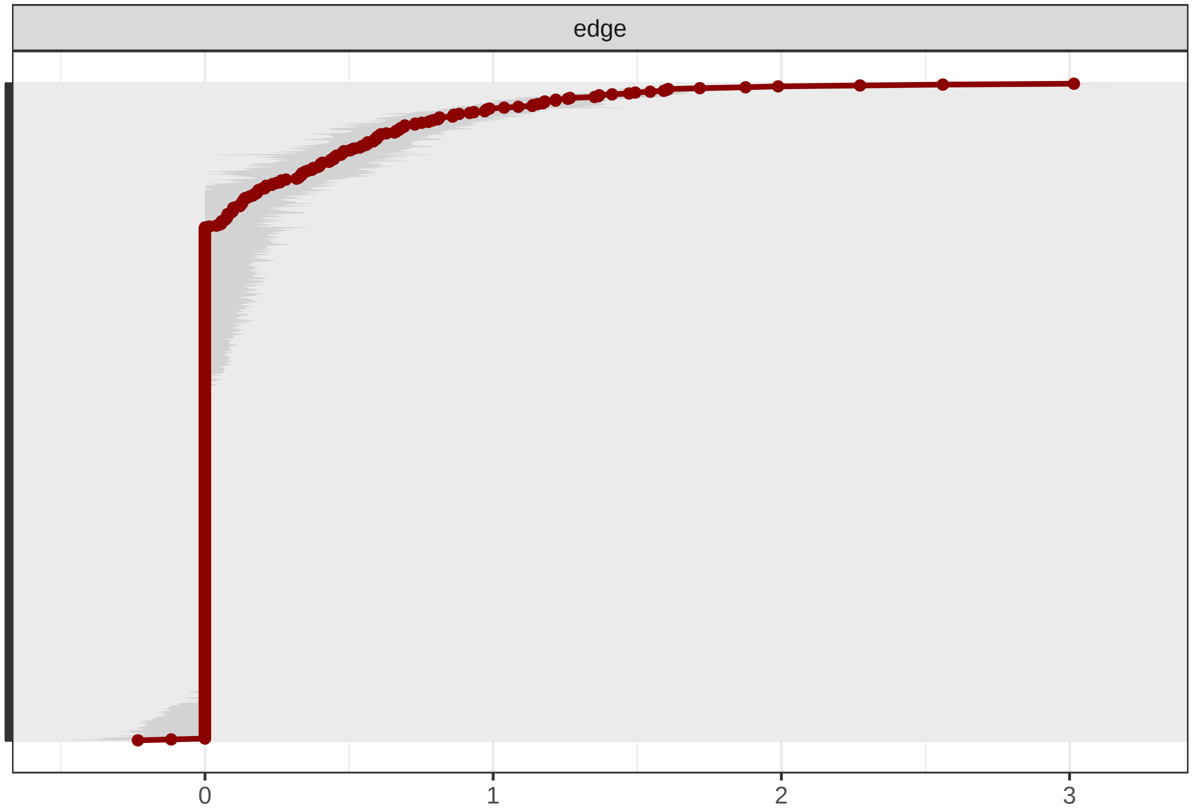
Col_big: Simulation Connectivity -> see NL vs Col

Col_big: Stability

Col_big: Stability Edges

```
set.seed(1)
# Col_Stabiliy <- bootnet(Col_binary_big, 1000, 'IsingFit')
# save(Col_Stabiliy, file = 'Col_Stabiliy.Rda')

load("Col_Stabiliy.Rda")
```

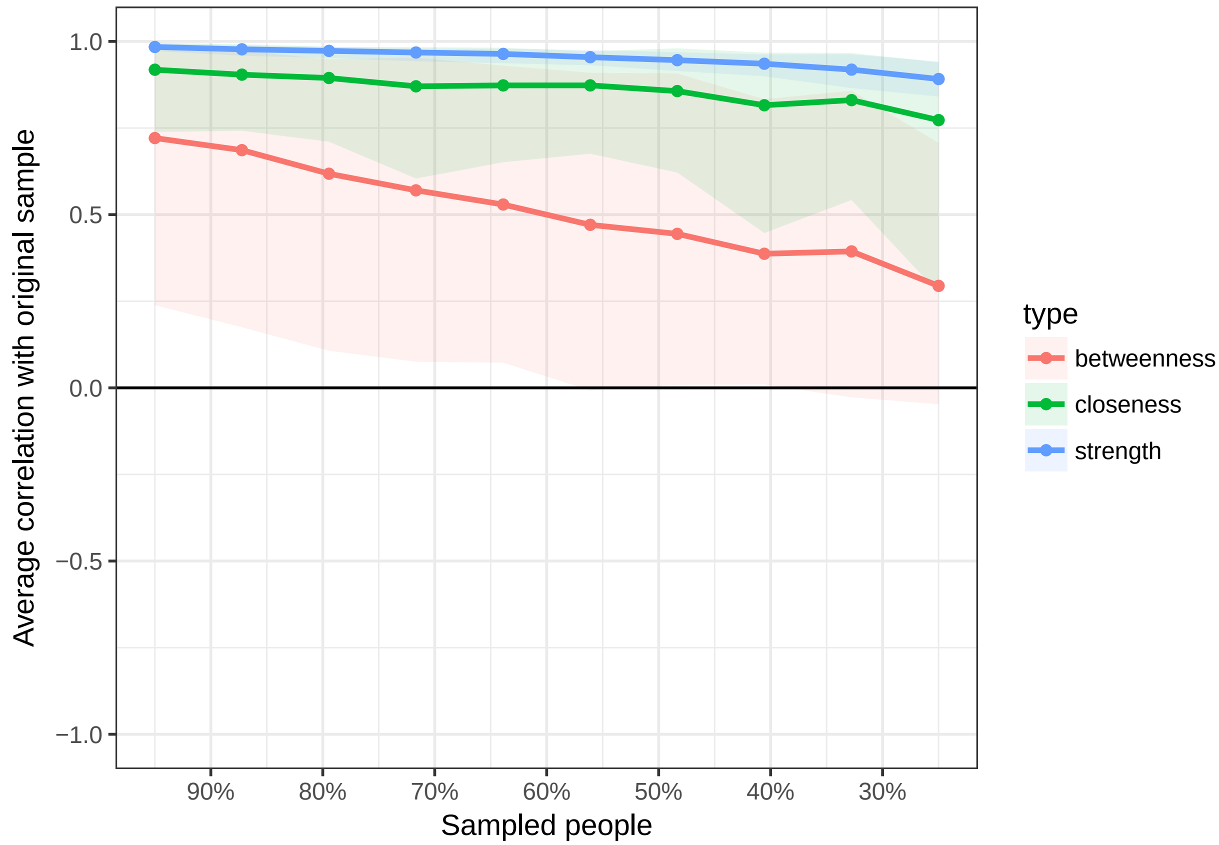
```r
# Figure S2 A tiff(filename = 'Figure S2 A.tiff', width = 6400, height =
# 6400, units = 'px', res = 800, compression = c('none'), bg = 'white', type
# = c('quartz'))
plot(Col_Stabiliy, order = "sample", labels = FALSE)
```




```r
# dev.off()
```

Col_big: Stability Centrality Indices

```r
# set.seed(1) Col_CentralityStability <- bootnet(Col_binary_big, 1000,
# 'IsingFit', 'person') save(Col_CentralityStability, file =
# 'Col_CentralityStability.Rda')

load("Col_CentralityStability.Rda")
# Figure S2 B tiff(filename = 'Figure S2 B.tiff', width = 6400, height =
# 6400, units = 'px', res = 800, compression = c('none'), bg = 'white', type
# = c('quartz'))
plot(Col_CentralityStability)
```




```r
# dev.off()

corStability(Col_CentralityStability)
```

```
## === Correlation Stability Analysis ===
##
## Sampling levels tested:
##    nPerson Drop%   n
## 1     1389  75.0 111
## 2     1821  67.2 106
## 3     2254  59.4  95
## 4     2686  51.7 103
## 5     3118  43.9  91
## 6     3550  36.1 101
## 7     3983  28.3  91
## 8     4415  20.6 101
## 9     4847  12.8  97
## 10    5279   5.0 104
##
## Maximum drop proportions to retain correlation of 0.7 in at least 95% of the samples:
##
## betweenness: 0
##   - For more accuracy, run bootnet(..., caseMin = 0, caseMax = 0.05)
##
## closeness: 0.206
##   - For more accuracy, run bootnet(..., caseMin = 0.128, caseMax = 0.283)
##
```

```
## strength: 0.75
##   - For more accuracy, run bootnet(..., caseMin = 0.672, caseMax = 1)
##
## Accuracy can also be increased by increasing both 'nBoots' and 'caseN'.
```

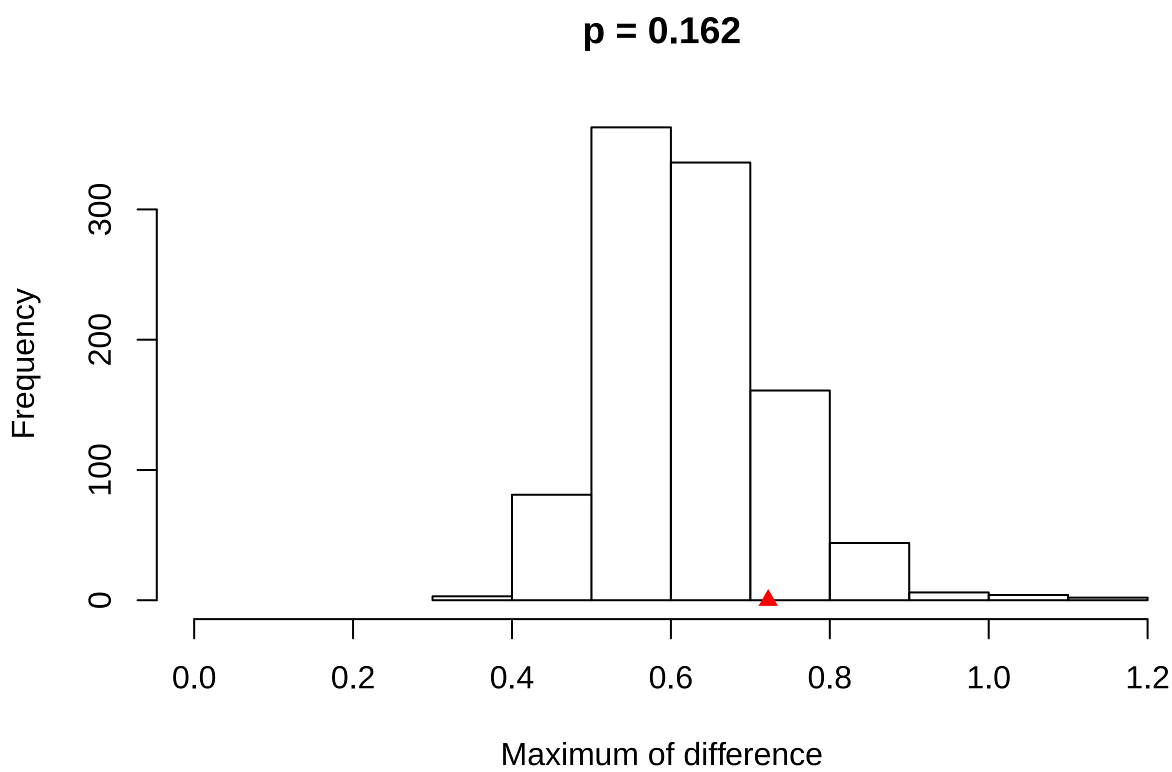
Col_big Boys vs. Girls

Col_big Boys vs. Girls_small Data

```
load("Col_binary_big.Rda")

# data frames boys vs. girls
Col_boys <- Col_binary_big[Col_binary_big$Gen == "0", ]
Col_girls <- Col_binary_big[Col_binary_big$Gen == "1", ]

Col_boys <- na.omit(Col_boys[, -c(1, 2, 42:44)])  # not in network: gender, SES etc
Col_girls <- na.omit(Col_girls[, -c(1, 2, 42:44)])  # not in network: gender, SES ect

# set.seed(1) Col_girls_small <- na.omit(Col_girls[sample(1:3151,
# 2406,replace = FALSE),]) # sample, so that boys and girls sample
# comparable save(Col_girls_small, file = 'Col_girls_small.Rda')

load("Col_girls_small.Rda")
```

Col_big Boys vs. Girls_small NCT -> structure invariance

```
###### network comparison Col_NCT_Gender_small <- NCT(Col_boys, Col_girls_small,
###### it = 1000, gamma = .25, binary.data = TRUE, paired = FALSE, test.edges =
###### TRUE, edges = 'all') save(Col_NCT_Gender_small,
###### file='Col_NCT_Gender_small.Rda')

load("Col_NCT_Gender_small.Rda")
# network structure invariant?
plot(Col_NCT_Gender_small, what = "network")
```




```
Col_NCT_Gender_small$nwinv.real
```

```
## [1] 0.7224716
```

```
Col_NCT_Gender_small$nwinv.pval
```

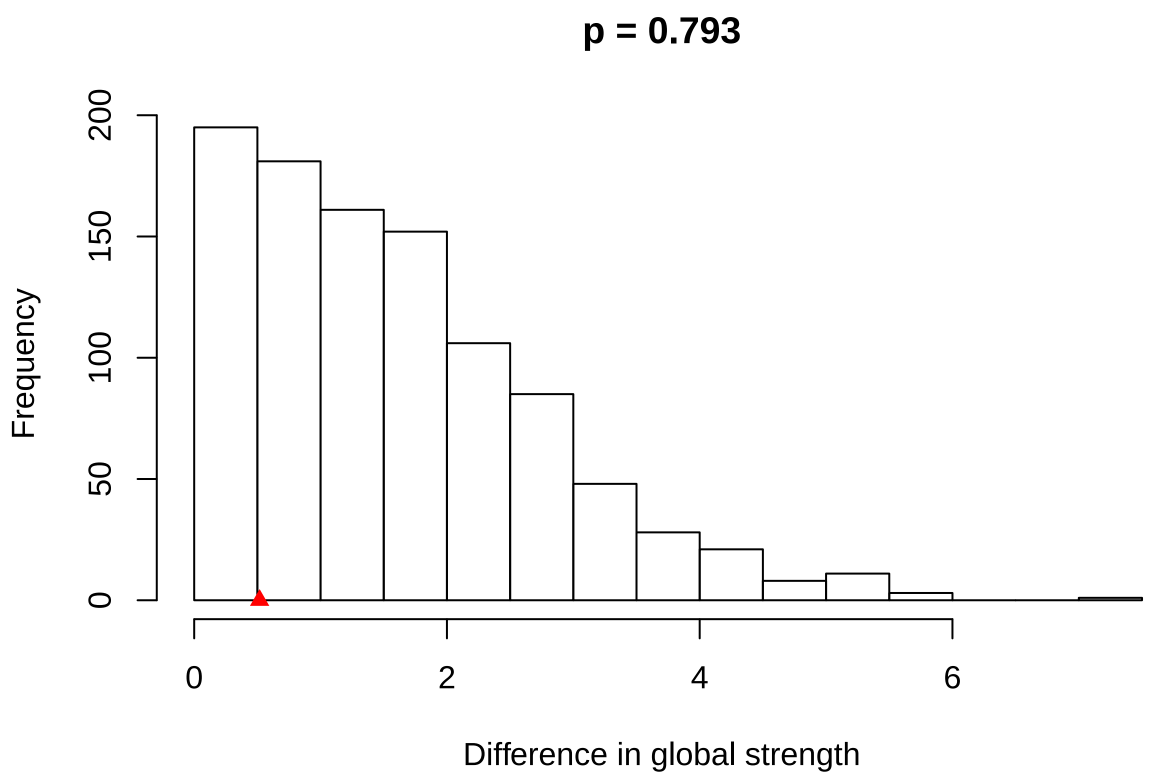
```
## [1] 0.162
```

Col_big Boys vs. Girls_small NCT -> specific edges differ? -> not done because networks not invariant

```
#p.edges <- Col_NCT_Gender_small$einv.pvals
```

Col_big Boys vs. Girls_small NCT -> global strength

```
# global strength
plot(Col_NCT_Gender_small, what = "strength")
```




```
Col_NCT_Gender_small$glstrinv.real
```

```
## [1] 0.5178127
```

```
Col_NCT_Gender_small$glstrinv.pval
```

```
## [1] 0.793
```

```
# global strength per network
Col_NCT_Gender_small$glstrinv.sep
```

```
## [1] 90.11899 89.60118
```

Col_big Low vs. high SES

Col_big Low vs. high SES Data

```
load("Col_binary_big.Rda")

# data frames low vs high SES
quantile(Col_binary_big$SES)
```

```
##      0%     25%     50%     75%    100%
## -4.2475 -1.3743 -0.6563  0.1266  2.0187
```

```
lowSES <- Col_binary_big[Col_binary_big$SES <= -1.3743, ]  # for quantiles: see data inspection
highSES <- Col_binary_big[Col_binary_big$SES >= 0.1266, ]

#
lowSES <- na.omit(lowSES[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age
```

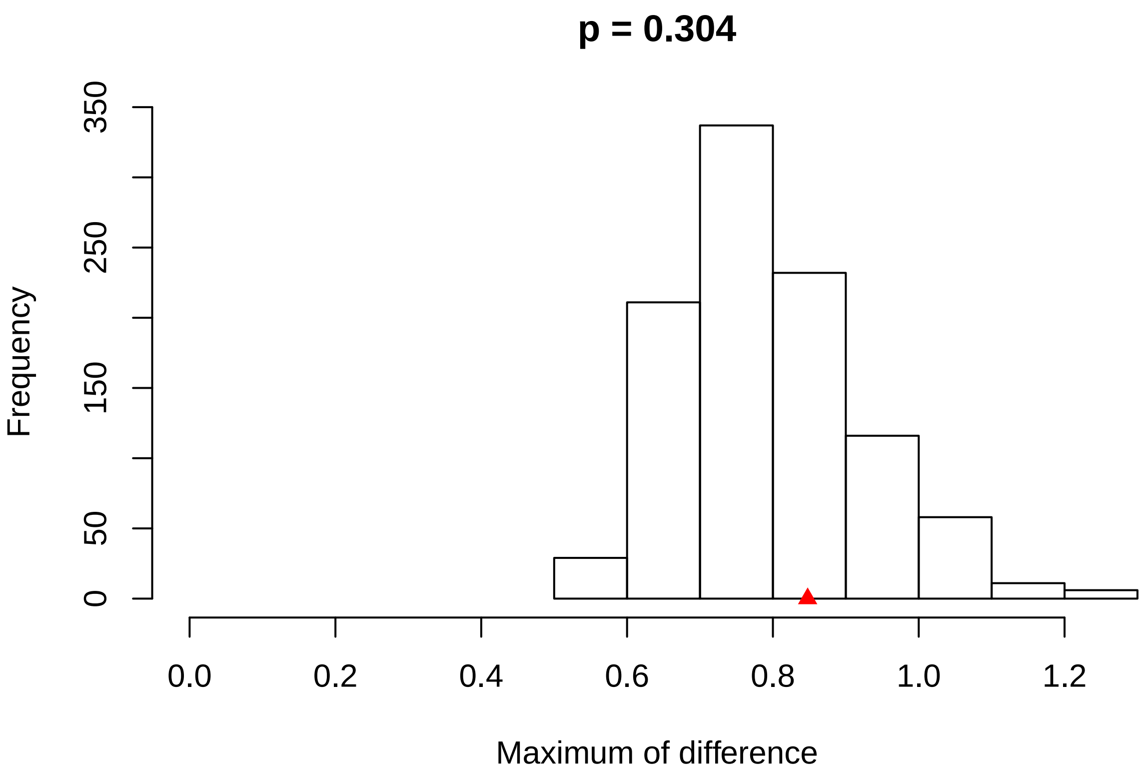
```r
highSES <- na.omit(highSES[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age
```

Col_big Low vs. high SES NCT

```r
# Col_NCT_SES <- NCT(lowSES, highSES, it = 1000, gamma = .25, binary.data =
# TRUE, paired = FALSE, test.edges = TRUE, edges = 'all') save(Col_NCT_SES,
# file='Col_NCT_SES.Rda')

load("Col_NCT_SES.Rda")
```

Col_big Low vs. high SES NCT -> structure invariance

```r
plot(Col_NCT_SES, what = "network")
```




```r
Col_NCT_SES$nwinv.real
```

```
## [1] 0.8476424
```

```r
Col_NCT_SES$nwinv.pval
```

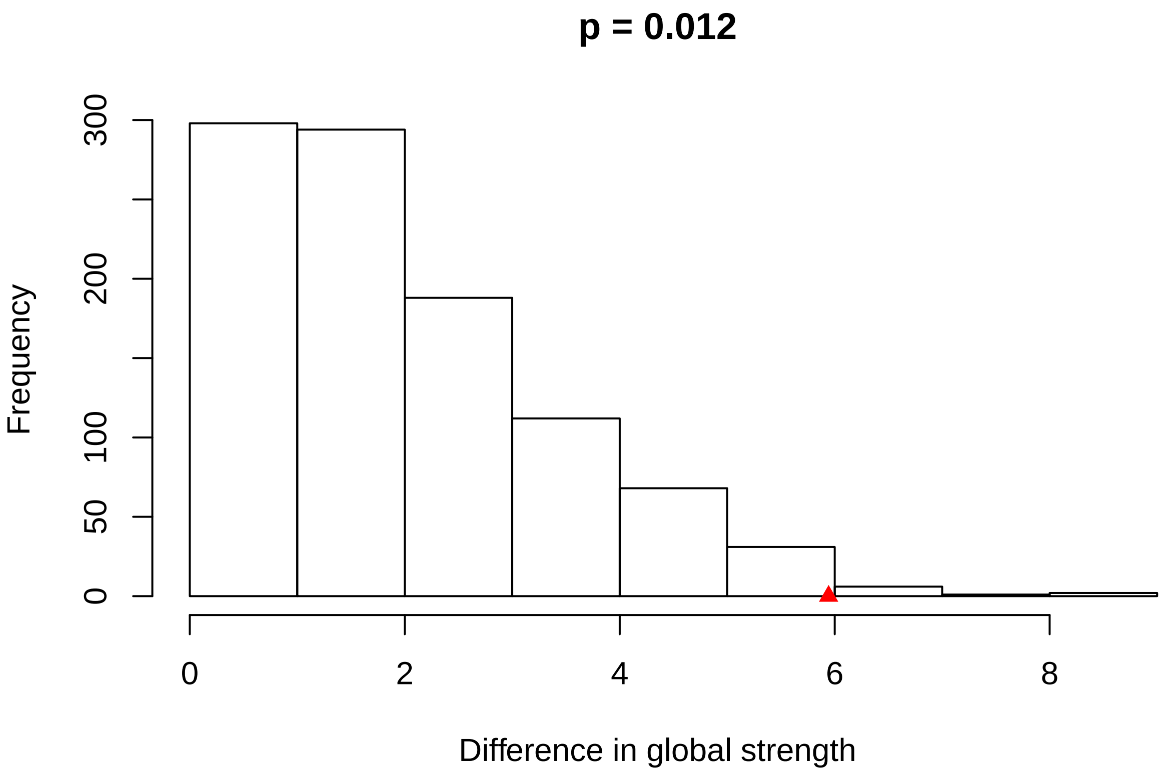
```
## [1] 0.304
```

Col_big Low vs. high SES NCT -> specific edges differ? -> not done because networks not invariant

```r
#p.edges <- Col_NCT_SES$einv.pvals #
#which(p.edges$`p-value`<0.05)
```

Col_big Low vs. high SES NCT -> global strength

```r
plot(Col_NCT_SES, what = "strength")
```




```r
Col_NCT_SES$glstrinv.real
```

```
## [1] 5.943545
```

```r
Col_NCT_SES$glstrinv.pval
```

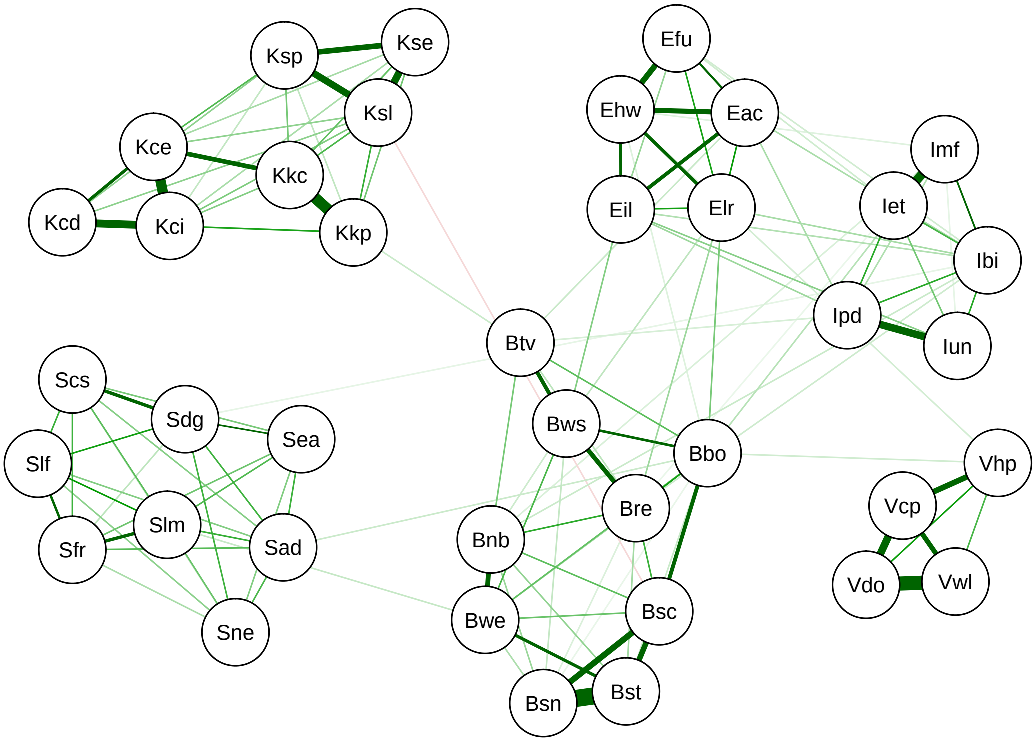
```
## [1] 0.012
```

```r
# global strength per network
Col_NCT_SES$glstrinv.sep
```

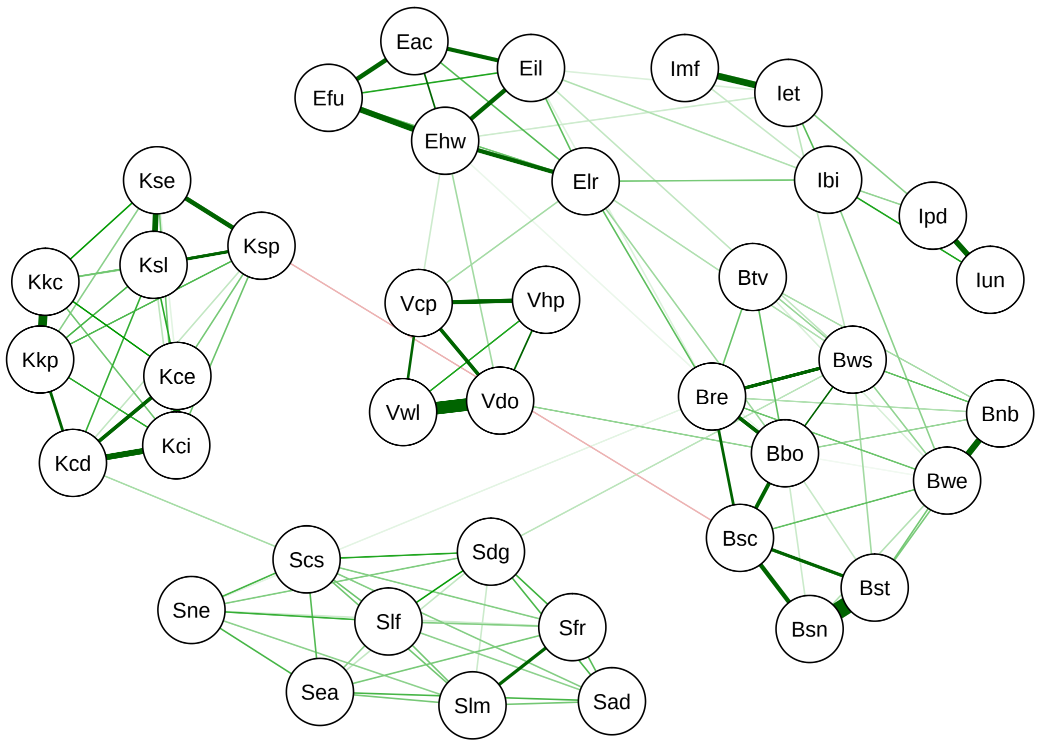
```
## [1] 90.05694 84.11339
```

Col_big Low vs. high SES network estimation (because NCT -> global strength sign. different)

```r
### Low SES
lowSES_fit <- IsingFit(lowSES)
```



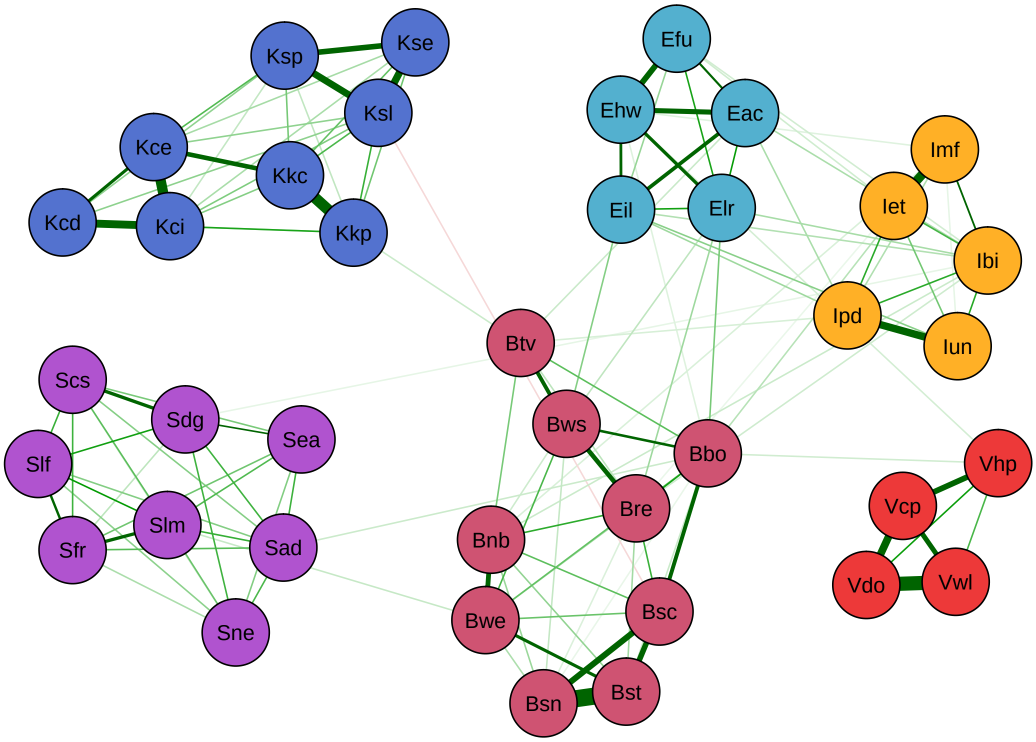

```
# Community detection and plotting
lowSES_igraph <- graph_from_adjacency_matrix(abs(lowSES_fit$weiadj), "undirected",
    weighted = TRUE, add.colnames = FALSE)
lowSES_Com <- cluster_walktrap(lowSES_igraph)
communities(lowSES_Com)
# smallworldness
SW_Index(lowSES_igraph)

### High SES
highSES_fit <- IsingFit(highSES)
```



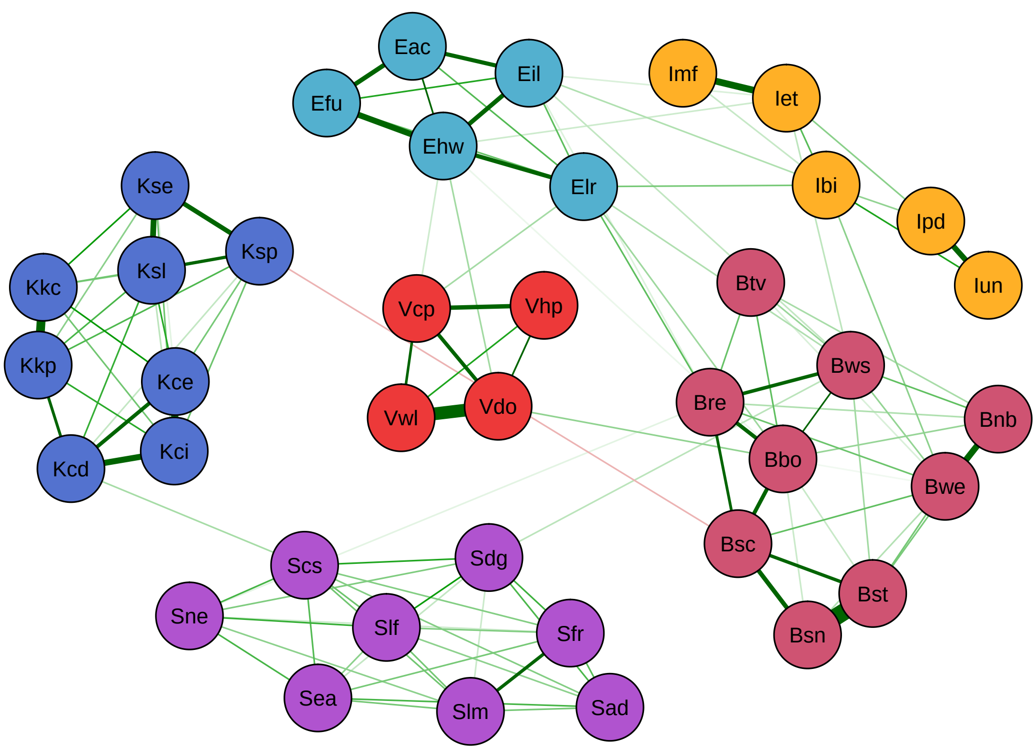

```
# Community detection
highSES_igraph <- graph_from_adjacency_matrix(abs(highSES_fit$weiadj), "undirected",
    weighted = TRUE, add.colnames = FALSE)
highSES_Com <- cluster_walktrap(highSES_igraph)
communities(highSES_Com)
# smallworldness
SW_Index(highSES_igraph)
```

Col_big Low vs. high SES Plots -> FIGURE 6 & 7

```
# Plots -----------------------------------------------------
group_col <- c("#53B0CF", "#FFB026", "#ED3939", "#cf5372", "#b053cf", "#5372cf")

# Figure 6 tiff(filename = 'Figure 6.tiff', width = 6400, height = 3200,
# units = 'px', res = 800, compression = c('none'), bg = 'white', type =
# c('quartz')) layout(t(1:2))
lowSES_graph <- qgraph(lowSES_fit$weiadj, layout = "spring", color = group_col,
    groups = groups_type, nodeNames = names, legend = FALSE, legend.mode = "style2",
    legend.cex = 0.3)
```

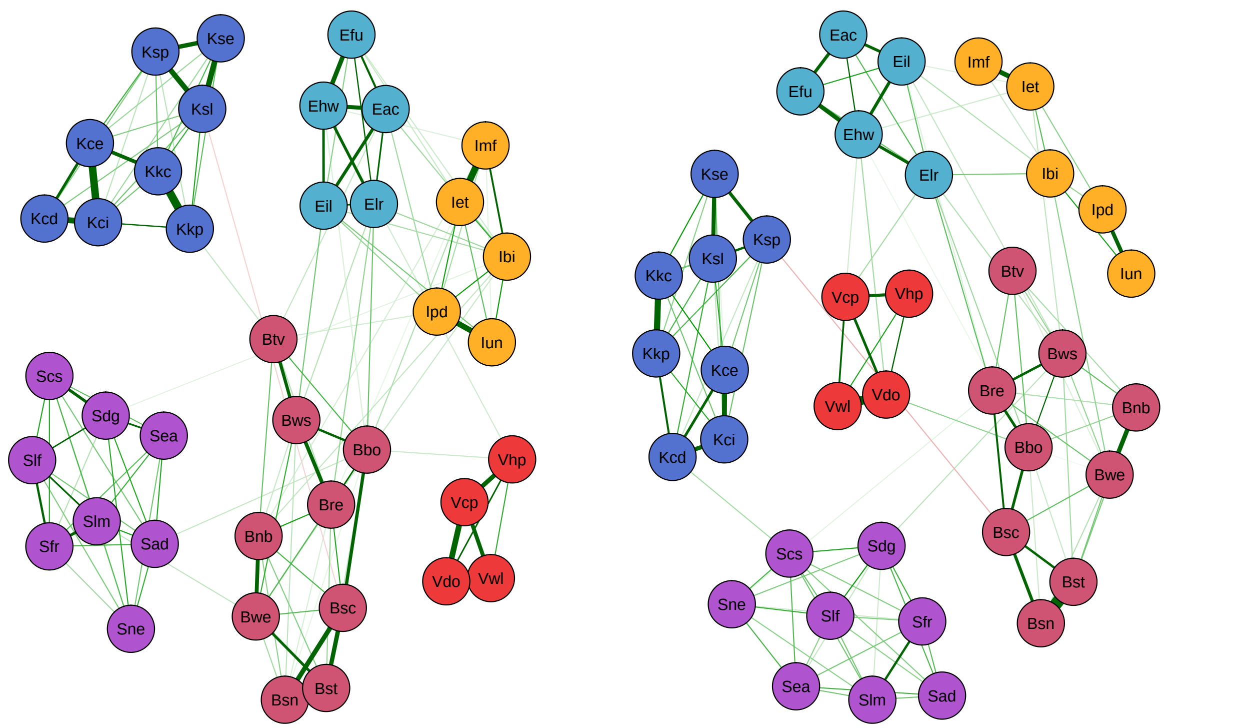



```r
highSES_graph <- qgraph(highSES_fit$weiadj, layout = "spring", color = group_col,
    groups = groups_type, nodeNames = names, legend = FALSE, legend.mode = "style2",
    legend.cex = 0.3)
```



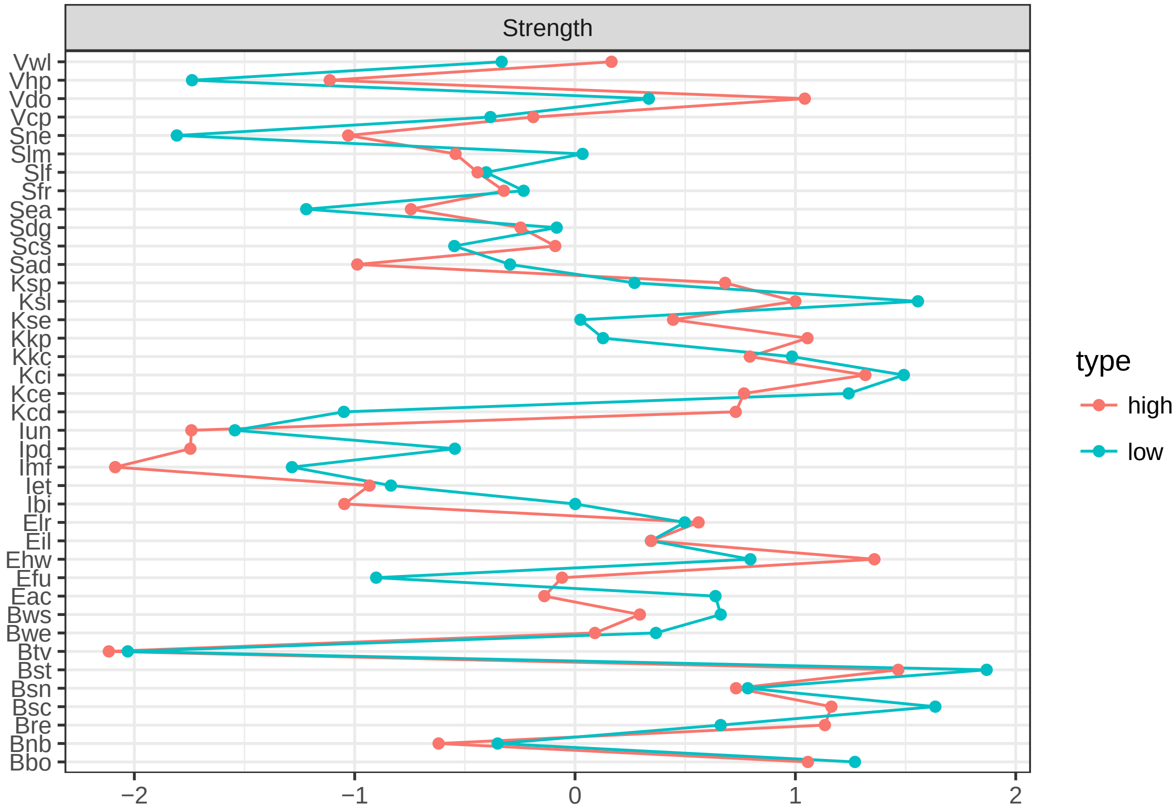

```r
# dev.off()

# community:
```

```r
group_col_comm_lowSES <- c("#cf5372", "#5372cf", "#FFB026", "#53B0CF", "#ED3939",
    "#b053cf")
group_col_comm_highSES <- c("#FFB026", "#cf5372", "#53B0CF", "#ED3939", "#5372cf",
    "#b053cf")

# Same as Figure 6 tiff(filename = 'Figure 6.1.tiff', width = 6400, height =
# 3200, units = 'px', res = 800, compression = c('none'), bg = 'white', type
# = c('quartz'))
layout(t(1:2))
qgraph(lowSES_fit$weiadj, layout = "spring", cut = 0.8, groups = communities(lowSES_Com),
    legend = FALSE, color = group_col_comm_lowSES)
qgraph(highSES_fit$weiadj, layout = "spring", groups = communities(highSES_Com),
    legend = FALSE, color = group_col_comm_highSES)
```




```r
# dev.off()

# Figure 7: centrality tiff(filename = 'Figure 7.tiff', width = 3200, height
# = 6400, units = 'px', res = 800, compression = c('none'), bg = 'white',
# type = c('quartz'))
centralityPlot(list(low = lowSES_graph, high = highSES_graph), include = "Strength",
    print = FALSE)
```

```
## Note: z-scores are shown on x-axis rather than raw centrality indices.
```




```r
# dev.off()
```

Col_big Academia vs. tecnica

Col_big Academia_small vs. tecnica Data

```r
load("Col_binary_big.Rda")

# data frames low vs high SES
academia <- Col_binary_big[Col_binary_big$Lev == "1", ]
tecnica <- Col_binary_big[Col_binary_big$Lev == "2", ]
academia <- na.omit(academia[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age
tecnica <- na.omit(tecnica[, -c(1, 2, 42:44)])  # not in network: ID, Gender, SES, Lev, Age

# subsample academia set.seed(1) academia_small <-
# na.omit(academia[sample(1:3775, 1782,replace = FALSE),])
# save(academia_small, file = 'Col_academia_small.Rda')

load("Col_academia_small.Rda")
```

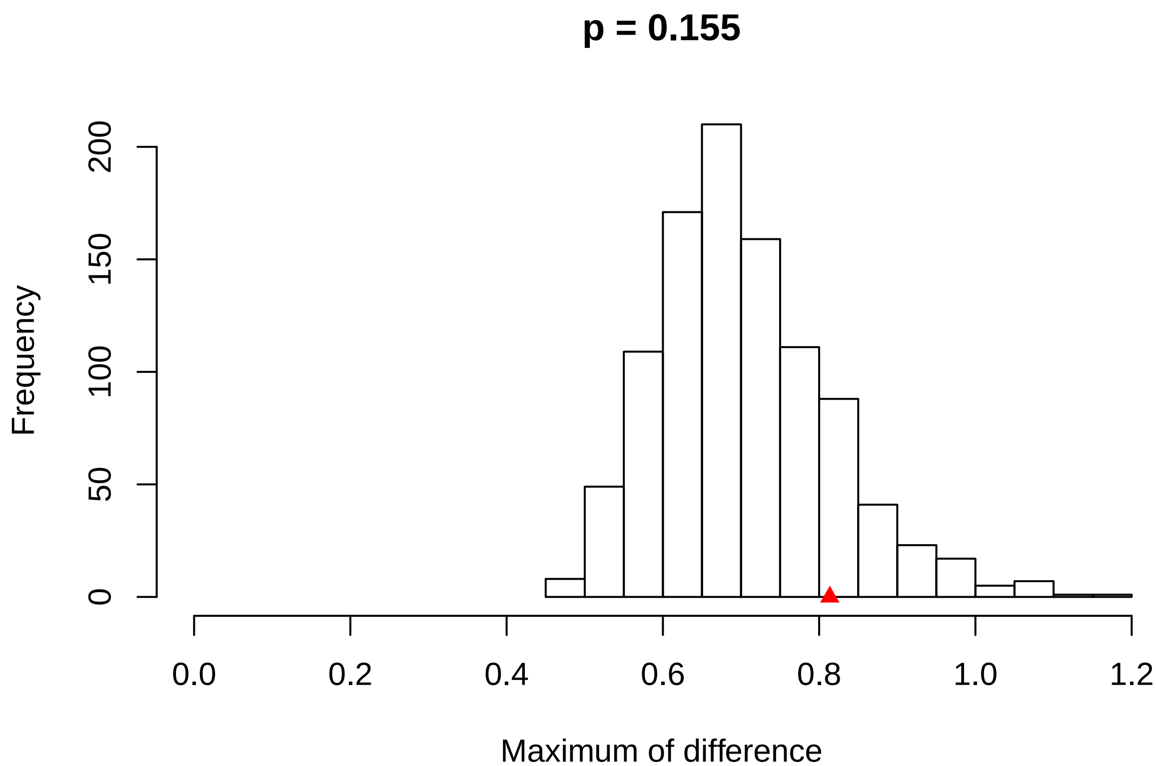
Col_big Academia_small vs. tecnica NCT

```r
# Col_NCT_AcaVsTec_small <- NCT(academia_small, tecnica, it = 1000, gamma =
# .25, binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges = 'all')
# save(Col_NCT_AcaVsTec_small, file='Col_NCT_AcaVsTec_small.Rda')
```

```r
load("Col_NCT_AcaVsTec_small.Rda")
```

Col_big Academia_small vs. tecnica NCT -> structure invariance

```r
plot(Col_NCT_AcaVsTec_small, what = "network")
```




```r
Col_NCT_AcaVsTec_small$nwinv.real
```

```
## [1] 0.8136828
```

```r
Col_NCT_AcaVsTec_small$nwinv.pval
```

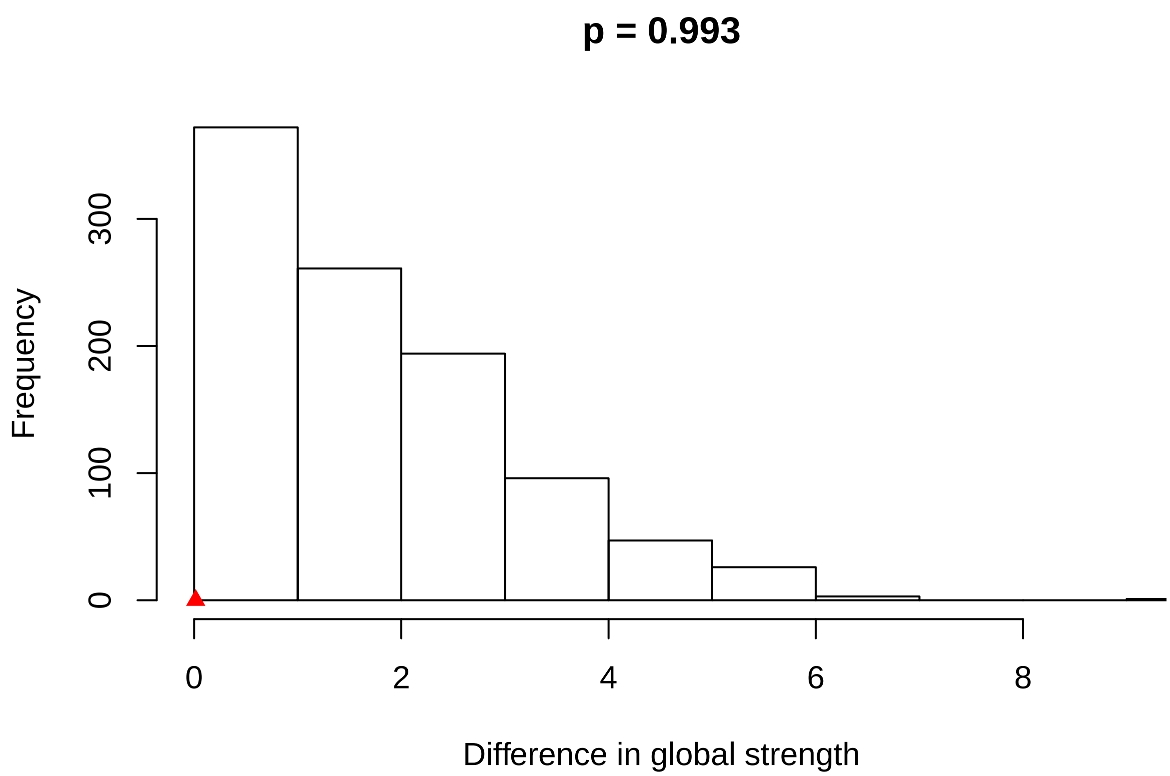
```
## [1] 0.155
```

Col_big Academia_small vs. tecnica NCT -> specific edges differ? -> not done because networks not invariant

```r
#p.edges <- Col_NCT_AcaVsTec_small$einv.pvals
#which(p.edges$`p-value`<0.05)
```

Col_big Academia_small vs. tecnica NCT -> global strength

```r
plot(Col_NCT_AcaVsTec_small, what = "strength")
```




```
Col_NCT_AcaVsTec_small$glstrinv.real
```

```
## [1] 0.01582891
```

```
Col_NCT_AcaVsTec_small$glstrinv.pval
```

```
## [1] 0.993
```

```
# global strength per network
Col_NCT_AcaVsTec_small$glstrinv.sep
```

```
## [1] 87.56199 87.54616
```

Col_big Lower vs. Upper secondary school

Col_big Lower vs. Upper_small Data preparation: Binarize lower

```
load("Col_nonBinary_big.Rda")
Col_binaryLower <- na.omit(Col_nonBinary_big[Col_nonBinary_big$Lev == "0", ])

### binarize variables Enjoyment:
Col_binaryLower$Efu[Col_binaryLower$Efu < 3] <- 0
Col_binaryLower$Efu[Col_binaryLower$Efu > 2] <- 1
Col_binaryLower$Elr[Col_binaryLower$Elr < 3] <- 0
Col_binaryLower$Elr[Col_binaryLower$Elr > 2] <- 1
Col_binaryLower$Ehw[Col_binaryLower$Ehw < 3] <- 0
Col_binaryLower$Ehw[Col_binaryLower$Ehw > 2] <- 1
Col_binaryLower$Eac[Col_binaryLower$Eac < 3] <- 0
Col_binaryLower$Eac[Col_binaryLower$Eac > 2] <- 1
Col_binaryLower$Eil[Col_binaryLower$Eil < 3] <- 0
```

```r
Col_binaryLower$Eil[Col_binaryLower$Eil > 2] <- 1

# interest:
Col_binaryLower$Ibi[Col_binaryLower$Ibi < 3] <- 0
Col_binaryLower$Ibi[Col_binaryLower$Ibi > 2] <- 1
Col_binaryLower$Imf[Col_binaryLower$Imf < 3] <- 0
Col_binaryLower$Imf[Col_binaryLower$Imf > 2] <- 1
Col_binaryLower$Iet[Col_binaryLower$Iet < 3] <- 0
Col_binaryLower$Iet[Col_binaryLower$Iet > 2] <- 1
Col_binaryLower$Iun[Col_binaryLower$Iun < 3] <- 0
Col_binaryLower$Iun[Col_binaryLower$Iun > 2] <- 1
Col_binaryLower$Ipd[Col_binaryLower$Ipd < 3] <- 0
Col_binaryLower$Ipd[Col_binaryLower$Ipd > 2] <- 1

# Value:
Col_binaryLower$Vwl[Col_binaryLower$Vwl < 3] <- 0
Col_binaryLower$Vwl[Col_binaryLower$Vwl > 2] <- 1
Col_binaryLower$Vdo[Col_binaryLower$Vdo < 3] <- 0
Col_binaryLower$Vdo[Col_binaryLower$Vdo > 2] <- 1
Col_binaryLower$Vcp[Col_binaryLower$Vcp < 3] <- 0
Col_binaryLower$Vcp[Col_binaryLower$Vcp > 2] <- 1
Col_binaryLower$Vhp[Col_binaryLower$Vhp < 3] <- 0
Col_binaryLower$Vhp[Col_binaryLower$Vhp > 2] <- 1

# Behavior:
Col_binaryLower$Btv[Col_binaryLower$Btv < 2] <- 0
Col_binaryLower$Btv[Col_binaryLower$Btv > 1] <- 1
Col_binaryLower$Bbo[Col_binaryLower$Bbo < 2] <- 0
Col_binaryLower$Bbo[Col_binaryLower$Bbo > 1] <- 1
Col_binaryLower$Bws[Col_binaryLower$Bws < 2] <- 0
Col_binaryLower$Bws[Col_binaryLower$Bws > 1] <- 1
Col_binaryLower$Bre[Col_binaryLower$Bre < 2] <- 0
Col_binaryLower$Bre[Col_binaryLower$Bre > 1] <- 1
Col_binaryLower$Bsc[Col_binaryLower$Bsc < 2] <- 0
Col_binaryLower$Bsc[Col_binaryLower$Bsc > 1] <- 1
Col_binaryLower$Bsn[Col_binaryLower$Bsn < 2] <- 0
Col_binaryLower$Bsn[Col_binaryLower$Bsn > 1] <- 1
Col_binaryLower$Bst[Col_binaryLower$Bst < 2] <- 0
Col_binaryLower$Bst[Col_binaryLower$Bst > 1] <- 1
Col_binaryLower$Bwe[Col_binaryLower$Bwe < 2] <- 0
Col_binaryLower$Bwe[Col_binaryLower$Bwe > 1] <- 1
Col_binaryLower$Bnb[Col_binaryLower$Bnb < 2] <- 0
Col_binaryLower$Bnb[Col_binaryLower$Bnb > 1] <- 1

# Self efficacy
Col_binaryLower$Sne[Col_binaryLower$Sne < 3] <- 0
Col_binaryLower$Sne[Col_binaryLower$Sne > 2] <- 1
Col_binaryLower$Sea[Col_binaryLower$Sea < 3] <- 0
Col_binaryLower$Sea[Col_binaryLower$Sea > 2] <- 1
Col_binaryLower$Sad[Col_binaryLower$Sad < 3] <- 0
Col_binaryLower$Sad[Col_binaryLower$Sad > 2] <- 1
Col_binaryLower$Sdg[Col_binaryLower$Sdg < 3] <- 0
Col_binaryLower$Sdg[Col_binaryLower$Sdg > 2] <- 1
```

```
Col_binaryLower$Scs[Col_binaryLower$Scs < 3] <- 0
Col_binaryLower$Scs[Col_binaryLower$Scs > 2] <- 1
Col_binaryLower$Slf[Col_binaryLower$Slf < 3] <- 0
Col_binaryLower$Slf[Col_binaryLower$Slf > 2] <- 1
Col_binaryLower$Slm[Col_binaryLower$Slm < 3] <- 0
Col_binaryLower$Slm[Col_binaryLower$Slm > 2] <- 1
Col_binaryLower$Sfr[Col_binaryLower$Sfr < 3] <- 0
Col_binaryLower$Sfr[Col_binaryLower$Sfr > 2] <- 1

# Knowledge -> Median
Col_binaryLower$Kce[Col_binaryLower$Kce <= median(Col_binaryLower$Kce)] <- 0
Col_binaryLower$Kce[Col_binaryLower$Kce > median(Col_binaryLower$Kce)] <- 1
Col_binaryLower$Kcd[Col_binaryLower$Kcd <= median(Col_binaryLower$Kcd)] <- 0
Col_binaryLower$Kcd[Col_binaryLower$Kcd > median(Col_binaryLower$Kcd)] <- 1
Col_binaryLower$Kci[Col_binaryLower$Kci <= median(Col_binaryLower$Kci)] <- 0
Col_binaryLower$Kci[Col_binaryLower$Kci > median(Col_binaryLower$Kci)] <- 1
Col_binaryLower$Kkc[Col_binaryLower$Kkc <= median(Col_binaryLower$Kkc)] <- 0
Col_binaryLower$Kkc[Col_binaryLower$Kkc > median(Col_binaryLower$Kkc)] <- 1
Col_binaryLower$Kkp[Col_binaryLower$Kkp <= median(Col_binaryLower$Kkp)] <- 0
Col_binaryLower$Kkp[Col_binaryLower$Kkp > median(Col_binaryLower$Kkp)] <- 1
Col_binaryLower$Ksp[Col_binaryLower$Ksp <= median(Col_binaryLower$Ksp)] <- 0
Col_binaryLower$Ksp[Col_binaryLower$Ksp > median(Col_binaryLower$Ksp)] <- 1
Col_binaryLower$Ksl[Col_binaryLower$Ksl <= median(Col_binaryLower$Ksl)] <- 0
Col_binaryLower$Ksl[Col_binaryLower$Ksl > median(Col_binaryLower$Ksl)] <- 1
Col_binaryLower$Kse[Col_binaryLower$Kse <= median(Col_binaryLower$Kse)] <- 0
Col_binaryLower$Kse[Col_binaryLower$Kse > median(Col_binaryLower$Kse)] <- 1

# Save
save(Col_binaryLower, file = "Col_binaryLower.Rda")
```

Col_big Lower vs. Upper_small Data

```
load("Col_binaryLower.Rda")
load("Col_binary_big.Rda")

Col_lower <- na.omit(Col_binaryLower[, -c(1, 2, 42:44)])
Col_upper <- na.omit(Col_binary_big[, -c(1, 2, 42:44)])

# subsample academia set.seed(1) Col_upper_small <-
# na.omit(Col_upper[sample(1:5557, 2728,replace = FALSE),]) save(Col_upper,
# file = 'Col_upper_small.Rda')

load("Col_upper_small.Rda")
```

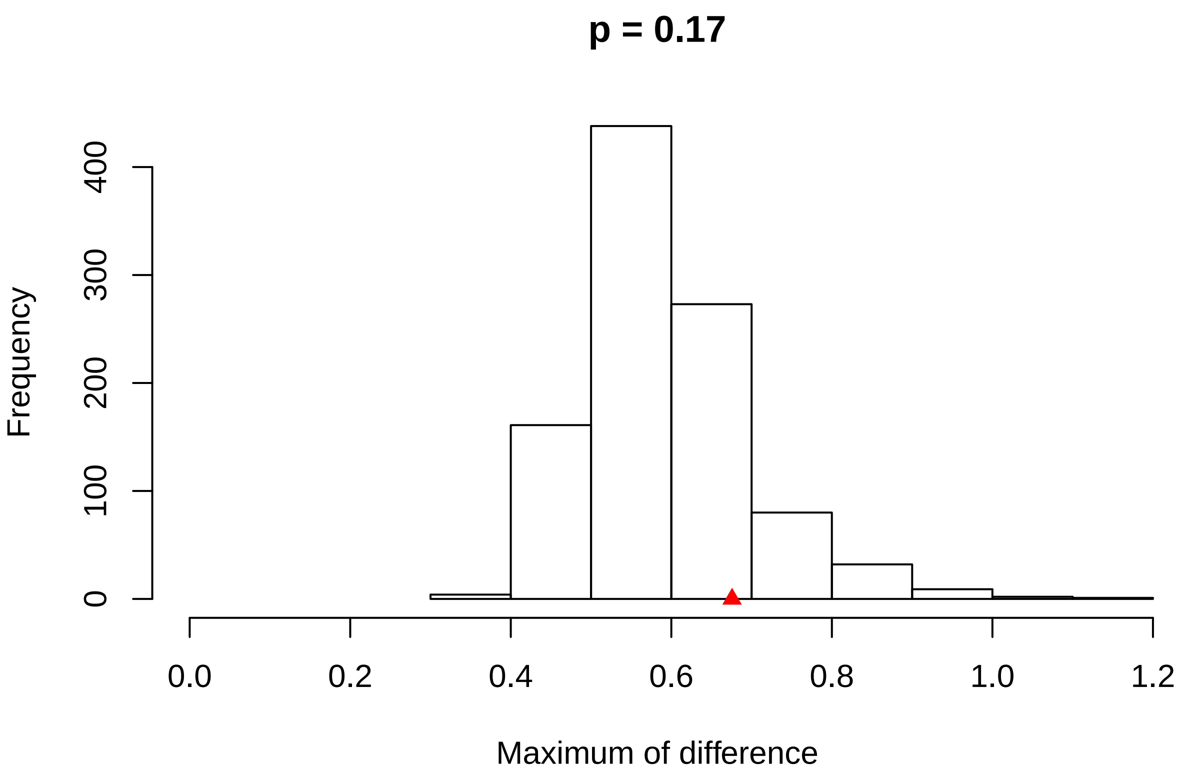
Col_big Lower vs. Upper_small NCT

```
# Col_NCT_LowVsUpper_small <- NCT(Col_lower, Col_upper_small, it = 1000,
# gamma = .25, binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges
# = 'all') save(Col_NCT_LowVsUpper_small,
# file='Col_NCT_LowVsUpper_small.Rda')
```

```r
load("Col_NCT_LowVsUpper_small.Rda")
```

Col_big lower vs. Upper_small NCT -> structure invariance

```r
plot(Col_NCT_LowVsUpper_small, what = "network")
```




```r
Col_NCT_LowVsUpper_small$nwinv.real
```

```
## [1] 0.6756684
```

```r
Col_NCT_LowVsUpper_small$nwinv.pval
```

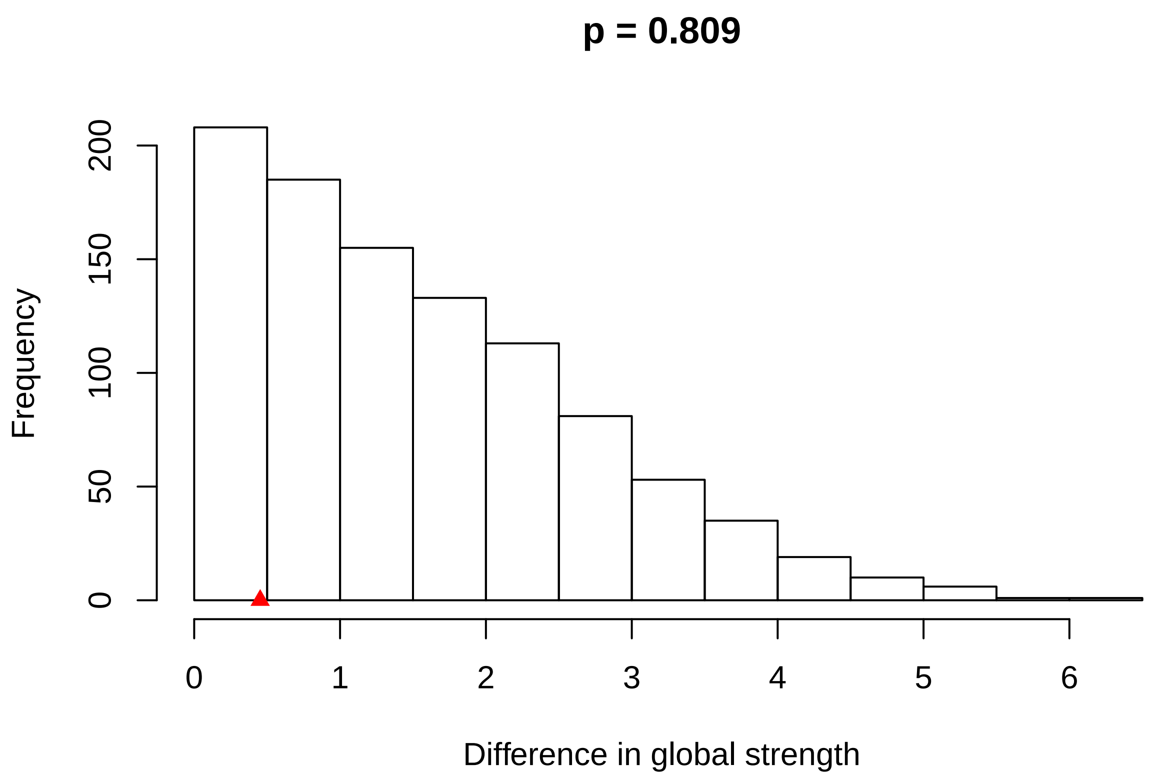
```
## [1] 0.17
```

Col_big lower vs. Upper_small NCT -> specific edges differ? -> not done because networks not invariant

```r
#p.edges <- Col_NCT_LowVsUpper_small$einv.pvals
#which(p.edges$`p-value`<0.05)
```

Col_big lower vs. Upper_small NCT -> global strength

```r
plot(Col_NCT_LowVsUpper_small, what = "strength")
```




```r
Col_NCT_LowVsUpper_small$glstrinv.real
```

```
## [1] 0.4524302
```

```r
Col_NCT_LowVsUpper_small$glstrinv.pval
```

```
## [1] 0.809
```

```r
# global strength per network
Col_NCT_LowVsUpper_small$glstrinv.sep
```

```
## [1] 92.81937 92.36694
```

NL vs. Col

NL vs. Col_small Data

```r
load("Col_binary_small.Rda")
Col_binary_small <- na.omit(Col_binary_small[-c(1,2,42:44)])

load("NL_binary.Rda")
NL_binary <- na.omit(NL_binary[-c(1,2,42:44)])
```

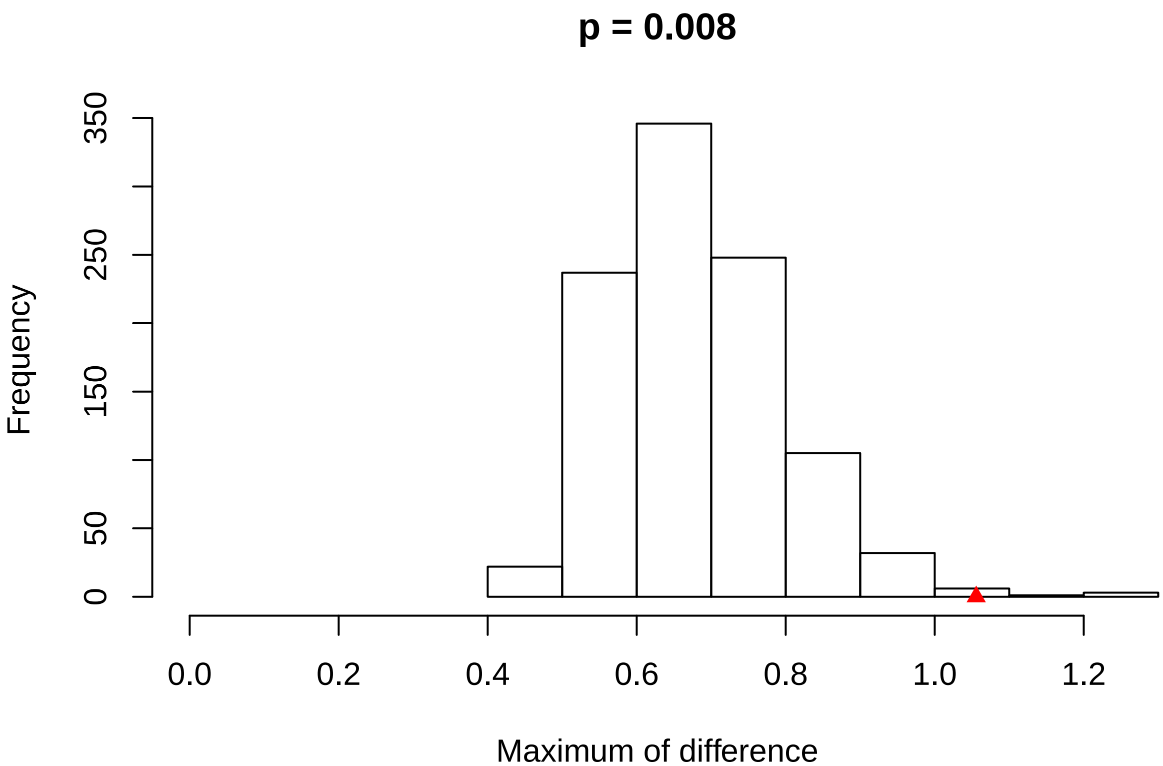
NL vs. Col_small NCT

```r
# NLvsCol_NCT_small <- NCT(NL_binary, Col_binary_small, it = 1000, gamma =
# .25, binary.data = TRUE, paired = FALSE, test.edges = TRUE, edges = 'all')
```

```r
# save(NLvsCol_NCT_small, file='NLvsCol_NCT_small.Rda')
load("NLvsCol_NCT_small.Rda")
```

NL vs. Col_small NCT -> invariance

```r
plot(NLvsCol_NCT_small, what = "network")
```




```r
NLvsCol_NCT_small$nwinv.real
```

```
## [1] 1.055705
```

```r
NLvsCol_NCT_small$nwinv.pval
```

```
## [1] 0.008
```

NL vs. Col_small NCT -> specific edges

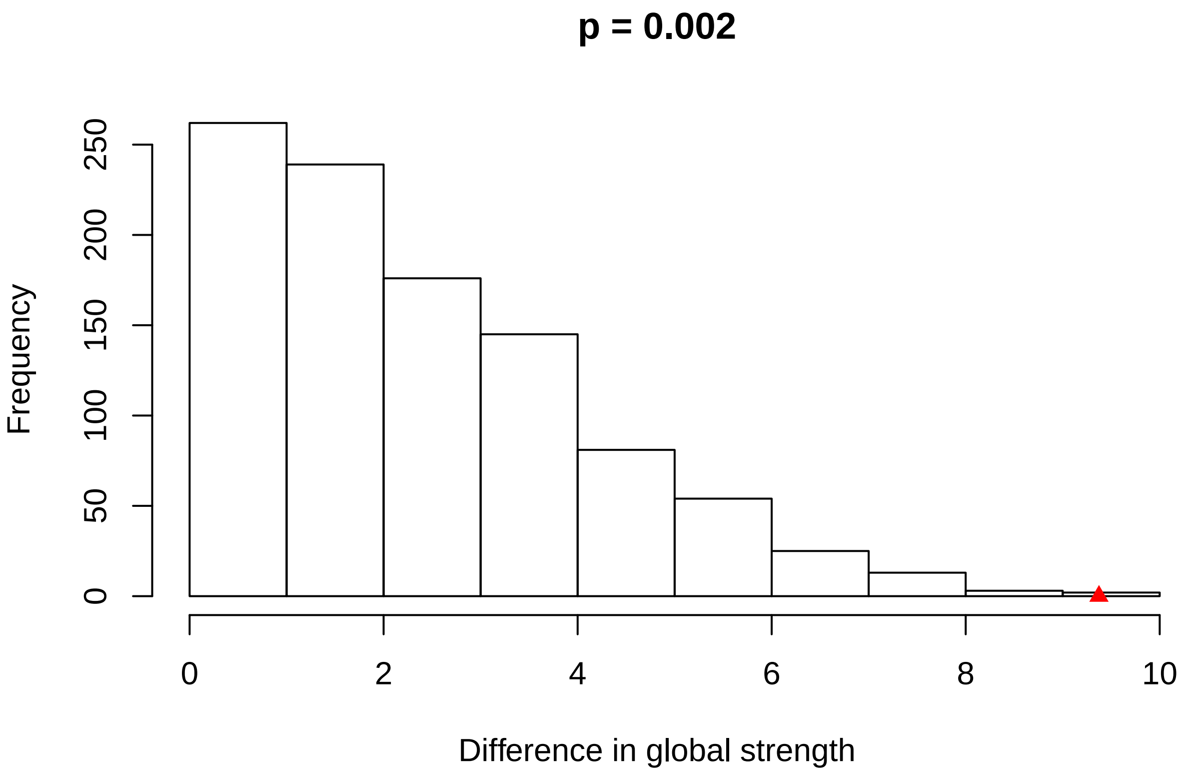
```r
p.edges <- NLvsCol_NCT_small$einv.pvals #
sign.edges <- p.edges[p.edges$`p-value`<0.05,]
sign.edges # shows which edges are significant
```

```
##       Var1 Var2 p-value
## 157    Efu  Eil       0
## 240    Ibi  Imf       0
## 359    Iet  Ipd       0
## 360    Iun  Ipd       0
## 440    Vwl  Vdo       0
## 520    Vcp  Vhp       0
```

```
## 640   Bbo  Bws        0
## 838   Bsc  Bwe        0
## 879   Bst  Bnb        0
## 907   Ipd  Sne        0
## 942   Ibi  Sea        0
## 945   Iun  Sea        0
## 951   Btv  Sea        0
## 981   Ibi  Sad        0
## 1176  Ibi  Sfr        0
## 1434  Slm  Ksp        0
## 1435  Sfr  Ksp        0
## 1517  Kkc  Kse        0
```

NL vs. Col_small NCT -> global strength

```r
# global strength
plot(NLvsCol_NCT_small, what = "strength")
```




```r
NLvsCol_NCT_small$glstrinv.real
```

```
## [1] 9.373845
```

```r
NLvsCol_NCT_small$glstrinv.pval
```

```
## [1] 0.002
```

```r
NLvsCol_NCT_small$glstrinv.sep
```

```
## [1] 98.01769 88.64384
```

NL vs. Col_big means -> TABLE 2

```r
load("NL_nonBinary.Rda")
NL <- na.omit(NL_nonBinary[, -c(1, 41:43)])
load("Col_nonBinary_big.Rda")  # non-binarized variables
Col <- na.omit(Col_nonBinary_big[, -c(1, 2, 42:44)])

# for t-tests
mKnoNL <- apply(NL[, c(32:39)], 1, mean)
sKnoNL <- apply(NL[, c(32:39)], 1, sd)
mKnoCol <- apply(Col[, c(32:39)], 1, mean)
sKnoCol <- apply(Col[, c(32:39)], 1, sd)
# Interest
mIntNL <- apply(NL[, c(6:10)], 1, mean)
sIntNL <- apply(NL[, c(6:10)], 1, sd)
mIntCol <- apply(Col[, c(6:10)], 1, mean)
sIntCol <- apply(Col[, c(6:10)], 1, sd)

# Value
mValNL <- apply(NL[, c(11:14)], 1, mean)
sValNL <- apply(NL[, c(11:14)], 1, sd)
mValCol <- apply(Col[, c(11:14)], 1, mean)
sValCol <- apply(Col[, c(11:14)], 1, sd)

# Enjoyment
mEnjNL <- apply(NL[, c(1:5)], 1, mean)
sEnjNL <- apply(NL[, c(1:5)], 1, sd)
mEnjCol <- apply(Col[, c(1:5)], 1, mean)
sEnjCol <- apply(Col[, c(1:5)], 1, sd)

# Behavior
mBehNL <- apply(NL[, c(15:23)], 1, mean)
sBehNL <- apply(NL[, c(15:23)], 1, sd)
mBehCol <- apply(Col[, c(15:23)], 1, mean)
sBehCol <- apply(Col[, c(15:23)], 1, sd)

# Self-Efficacy
mSelNL <- apply(NL[, c(24:31)], 2, mean)
sSelNL <- apply(NL[, c(24:31)], 2, sd)
mSelCol <- apply(Col[, c(24:31)], 2, mean)
sSelCol <- apply(Col[, c(24:31)], 2, sd)

# t-tests -> TABLE 2 knowledge
t.test(mKnoNL, mKnoCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mKnoNL and mKnoCol
## t = 78.728, df = 4207.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  129.8488 136.4811
## sample estimates:
```

```
## mean of x mean of y
##  588.1523  454.9874
```

```r
# Interest
t.test(mIntNL, mIntCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mIntNL and mIntCol
## t = -20.015, df = 3673.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3760585 -0.3089567
## sample estimates:
## mean of x mean of y
##  2.526163  2.868670
```

```r
# Value
t.test(mValNL, mValCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mValNL and mValCol
## t = -17.875, df = 3083.4, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4427916 -0.3552512
## sample estimates:
## mean of x mean of y
##  2.534171  2.933192
```

```r
# Enjoyment
t.test(mEnjNL, mEnjCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mEnjNL and mEnjCol
## t = -23.075, df = 3256.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4958777 -0.4182082
## sample estimates:
## mean of x mean of y
##  2.361672  2.818715
```

```r
# Behavior
t.test(mBehNL, mBehCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mBehNL and mBehCol
## t = -35.377, df = 6014.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
```

```
## 95 percent confidence interval:
##  -0.4762683 -0.4262565
## sample estimates:
## mean of x mean of y
##  1.377642  1.828904
```

```r
# Self-Efficacy
t.test(mSelNL, mSelCol, var.equal = FALSE)  #perform t-test gender difference
```

```
##
##  Welch Two Sample t-test
##
## data:  mSelNL and mSelCol
## t = 1.1583, df = 10.047, p-value = 0.2735
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1156037  0.3662657
## sample estimates:
## mean of x mean of y
##  2.787694  2.662363
```

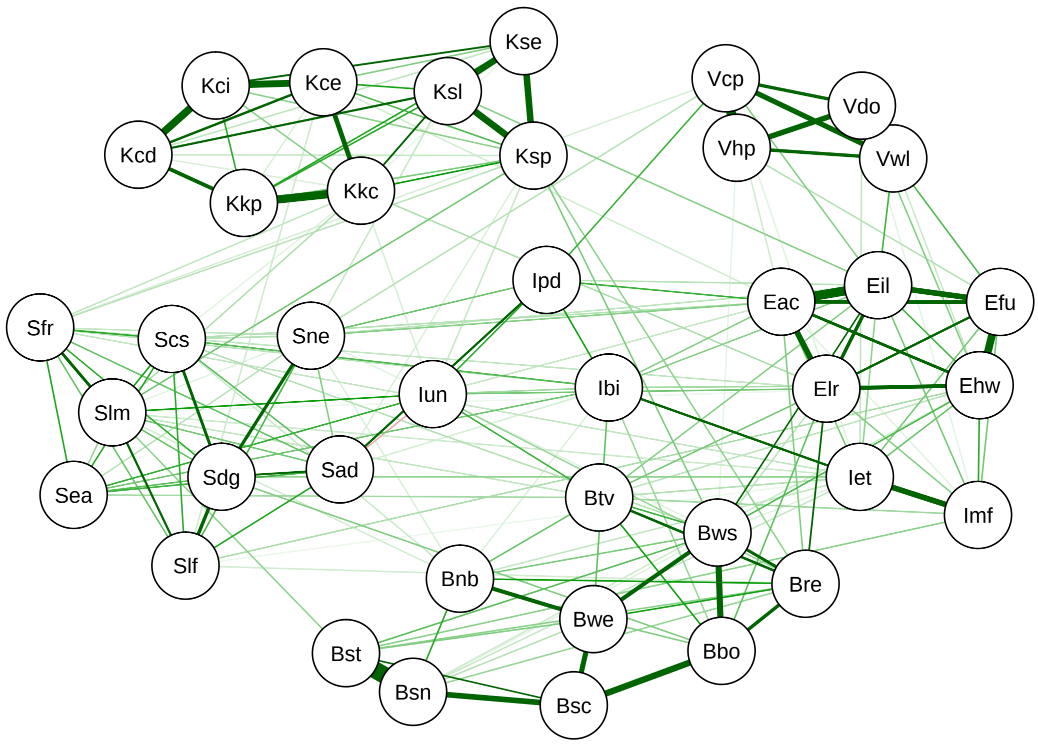
NL vs. Col_big Plotting

NL vs. Col_big Data

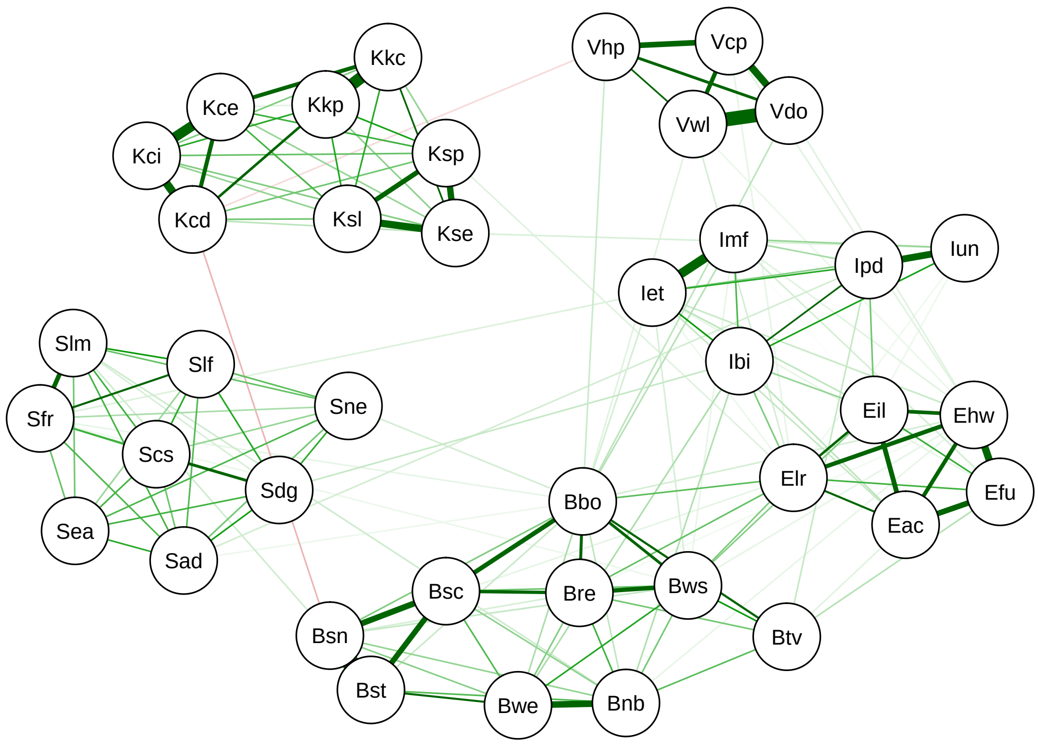
```r
load("Col_binary_big.Rda")
Col_binary_big <- na.omit(Col_binary_big[-c(1,2,42:44)])

load("NL_binary.Rda")
NL_binary <- na.omit(NL_binary[-c(1,2,42:44)])
```

NL vs. Col_big Network estimation

```r
# NL
NL_fit <- IsingFit(NL_binary)
```




```
# Community detection and plotting
NL_igraph <- graph_from_adjacency_matrix(abs(NL_fit$weiadj), "undirected", weighted = TRUE,
    add.colnames = FALSE)
NL_Com <- cluster_walktrap(NL_igraph)
communities(NL_Com)

#################### Colombia

Col_fit <- IsingFit(Col_binary_big)
```

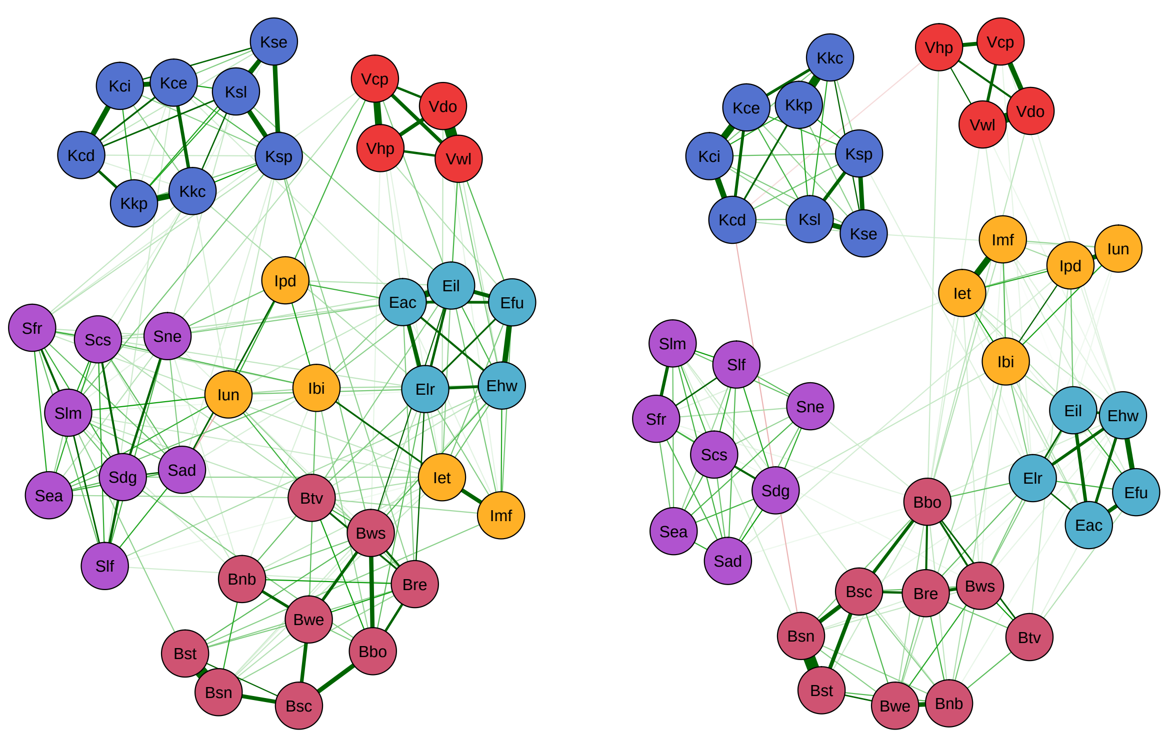



```
# Community detection
Col_igraph <- graph_from_adjacency_matrix(abs(Col_fit$weiadj), "undirected",
    weighted = TRUE, add.colnames = FALSE)
Col_Com <- cluster_walktrap(Col_igraph)
communities(Col_Com)
```

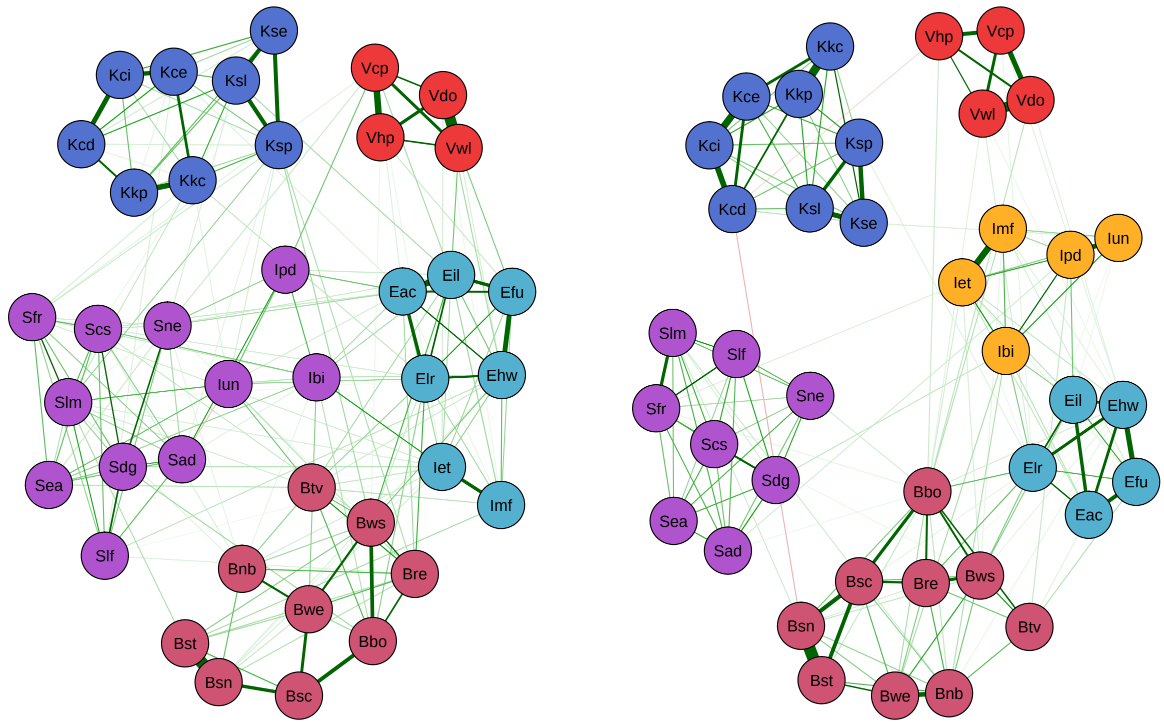
NL vs. Col_big Figures

```
# Plots -------------------------------------------------------
group_col <- c("#53B0CF", "#FFB026" ,"#ED3939", "#cf5372", "#b053cf", "#5372cf")

# Figure 2
#tiff(filename = "Figure 2.tiff",
#     width = 6400, height = 3200, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
layout(t(1:2))
Nl_graph <- qgraph(NL_fit$weiadj,
                   layout = 'spring',
                   color = group_col,
                   groups = groups_type,
                   nodeNames = names,
                   legend = FALSE,
                   legend.mode="style2",
                   legend.cex=.3)
Col_graph <- qgraph(Col_fit$weiadj,
                    layout = 'spring',
                    color = group_col,
```

```
                          groups = groups_type,
                          nodeNames = names,
                          legend = FALSE,
                          legend.mode="style2",
                          legend.cex=.3)
```

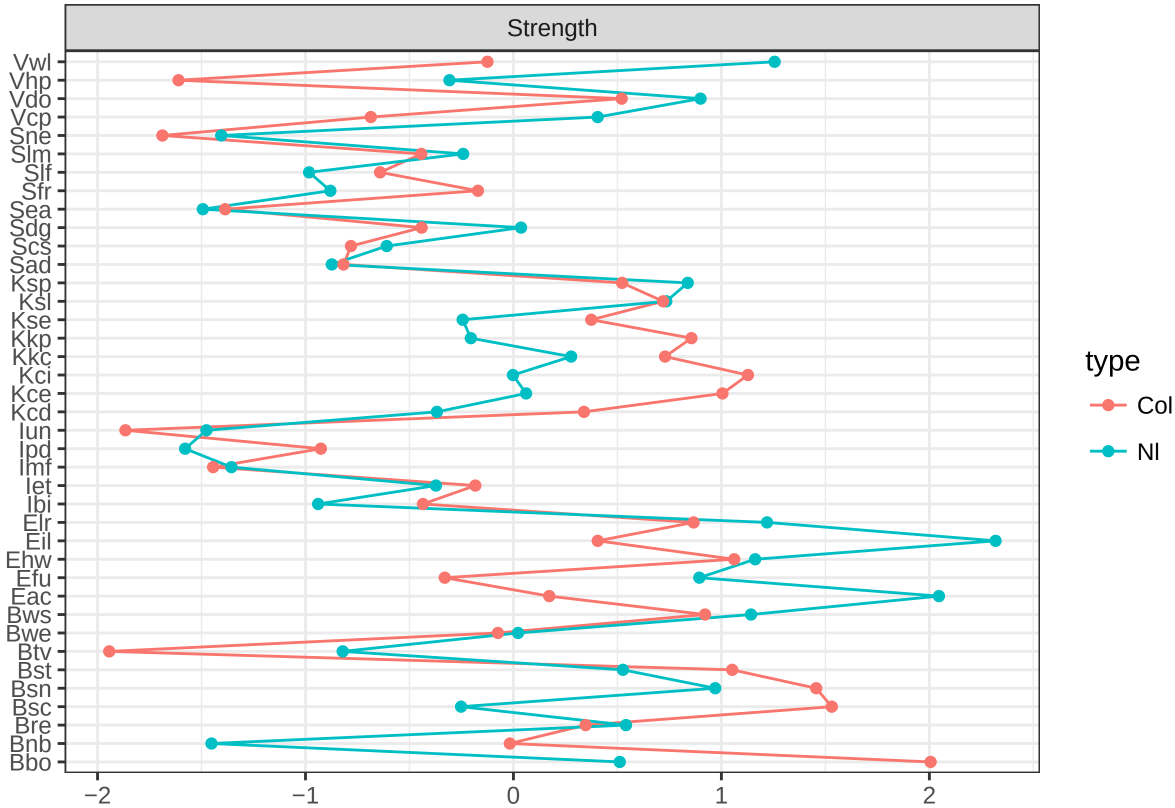



```
#dev.off()


# Figure 3: Community detection
group_col_comm_NL <- c("#cf5372", "#b053cf","#5372cf", "#53B0CF","#ED3939" )
group_col_comm_Col <- c("#cf5372", "#53B0CF", "#FFB026", "#5372cf", "#b053cf","#ED3939")

#tiff(filename = "Figure 3.tiff",
#     width = 6400, height = 3200, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
layout(t(1:2))
qgraph(NL_fit$weiadj, layout = 'spring', cut = .8,
       groups = communities(NL_Com), legend = FALSE,
       color = group_col_comm_NL)
qgraph(Col_fit$weiadj, layout = 'spring',
       groups = communities(Col_Com), legend = FALSE,
       color = group_col_comm_Col)
```




```r
#dev.off()

# Figure 4: centrality:
#tiff(filename = "Figure 4.tiff",
#    width = 3200, height = 6400, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
centralityPlot(list(Nl = Nl_graph, Col = Col_graph), print = FALSE, theme_bw = TRUE, include = "Strengt
```

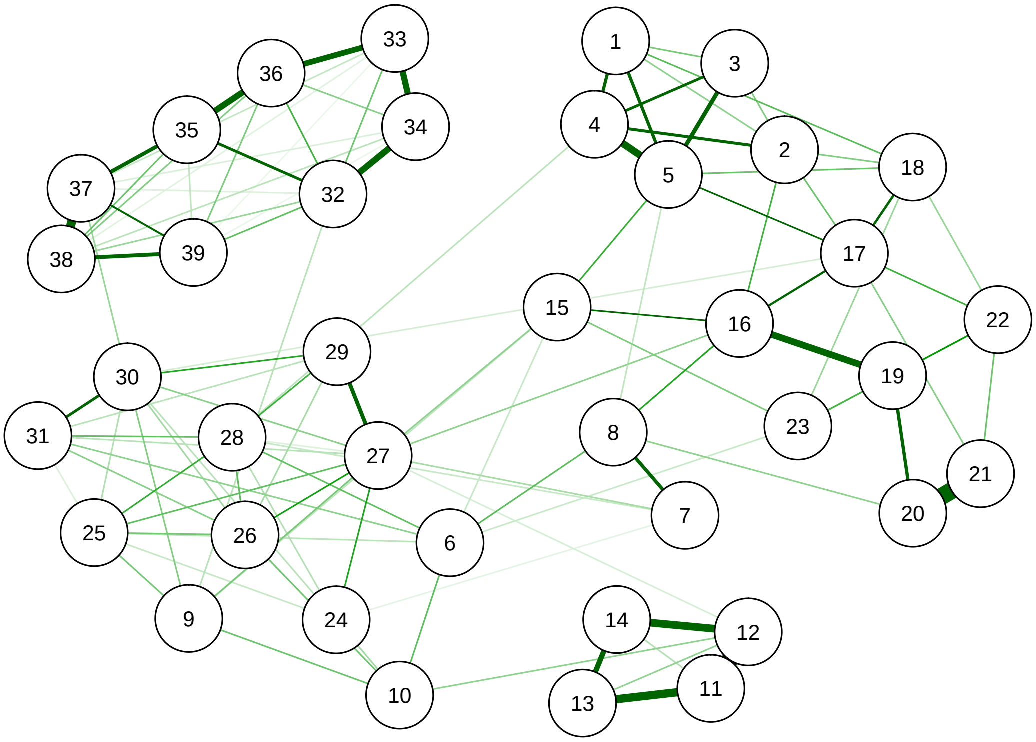
```
## Note: z-scores are shown on x-axis rather than raw centrality indices.
```



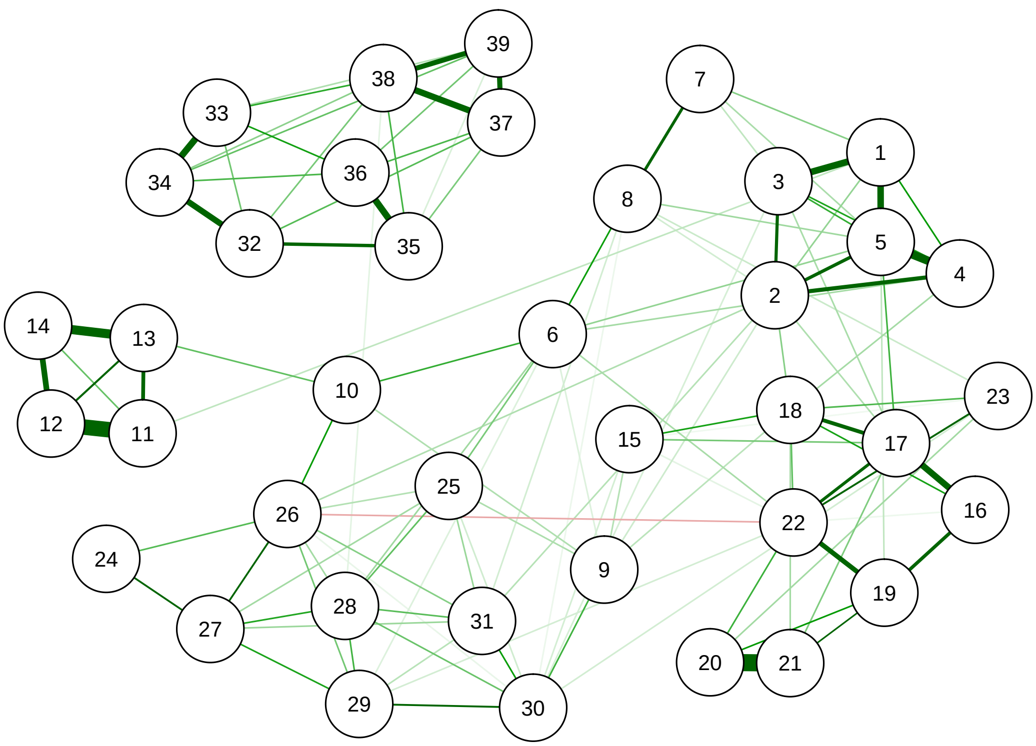

```
#dev.off()
```

NL vs. Col_big Connectivity Simulation plotting

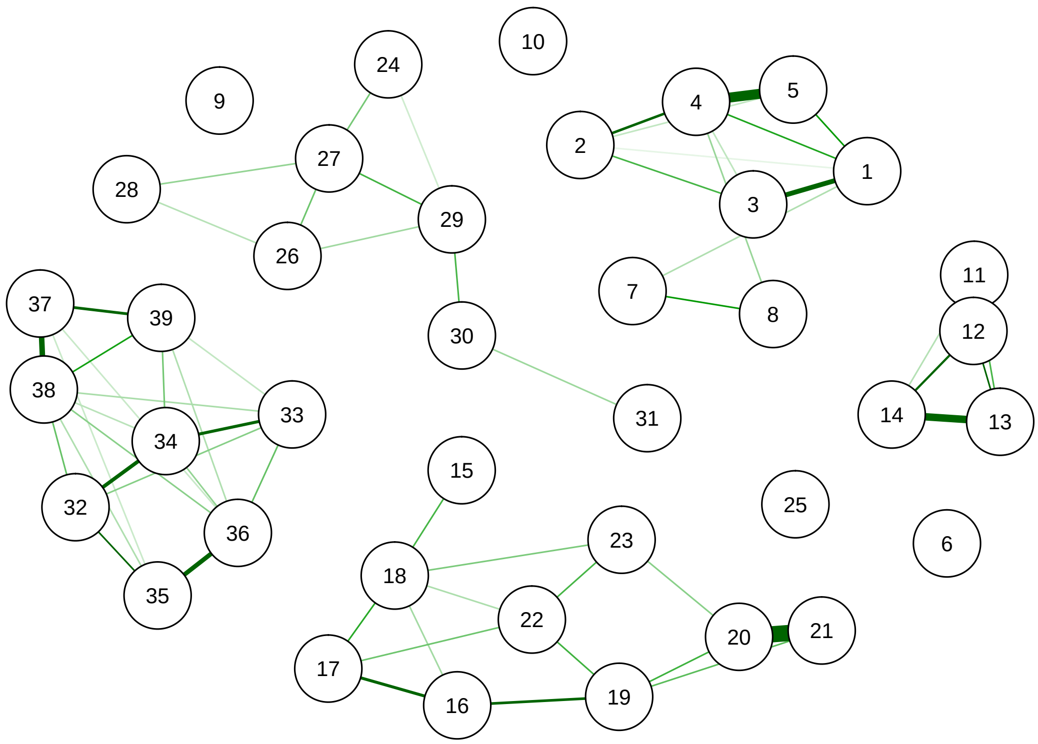
```
#################### Simulation input NL
NL_SimInput <- LinTransform(NL_fit$weiadj, NL_fit$thresholds)
#################### Connectivity simulation
set.seed(1)
NL_HighTemp <- IsingSampler(1000, NL_SimInput $ graph, rep(0,39), 1.8,
                            responses = c(-1L,1L))
NL_MidTemp <- IsingSampler(1000, NL_SimInput $ graph, rep(0,39), 1.2,
                           responses = c(-1L,1L))
NL_LowTemp <- IsingSampler(1000, NL_SimInput $ graph, rep(0,39), .6,
                           responses = c(-1L,1L))
# the simulated data contains -1;+1 responses. To estimate networks from the
# simulated data using IsingFit, 0s have to be assigned -1 responses
NL_HighTempResc <- NL_HighTemp
NL_HighTempResc[NL_HighTempResc == -1] <- 0

NL_MidTempResc <- NL_MidTemp
NL_MidTempResc[NL_MidTempResc == -1] <- 0

NL_LowTempResc <- NL_LowTemp
NL_LowTempResc[NL_LowTempResc == -1] <- 0
#these commands fit networks on the simulated data
NL_HighTempFit <- IsingFit(NL_HighTempResc)
```



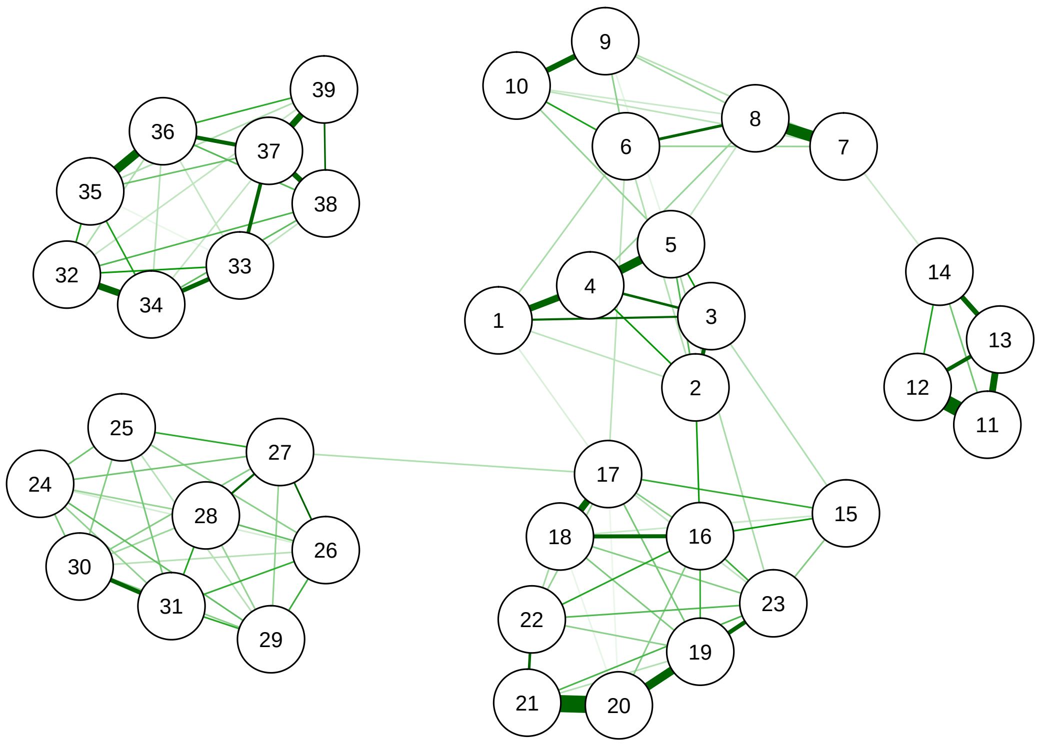
```
NL_MidTempFit <- IsingFit(NL_MidTempResc)
```



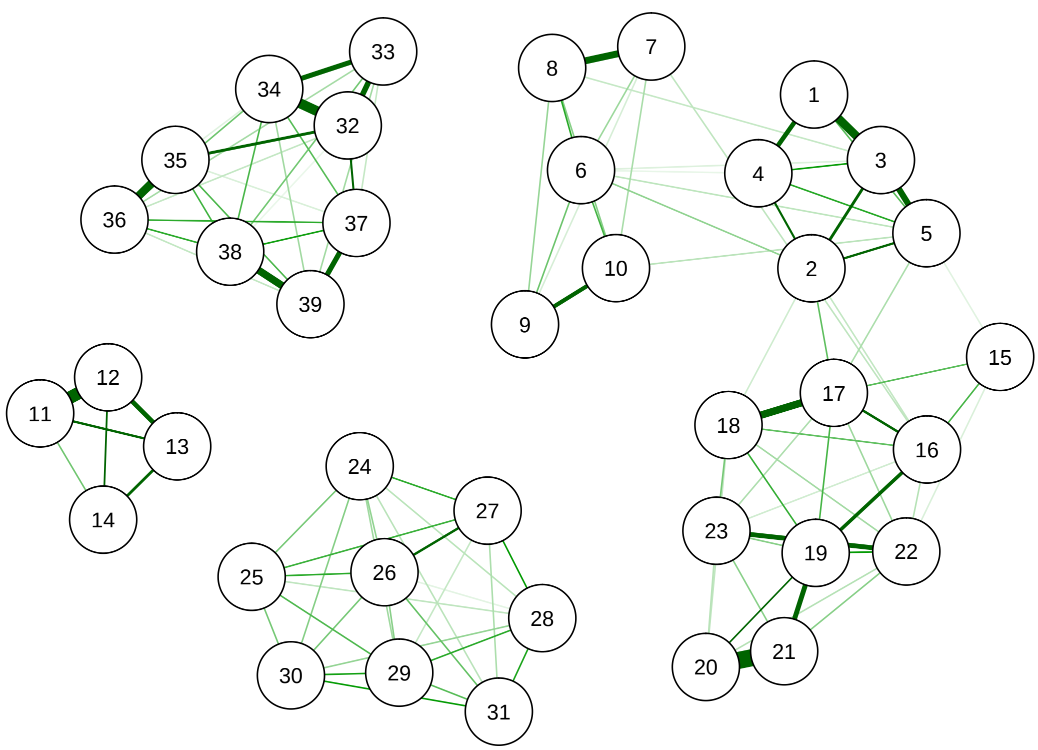
```
NL_LowTempFit <- IsingFit(NL_LowTempResc)
```



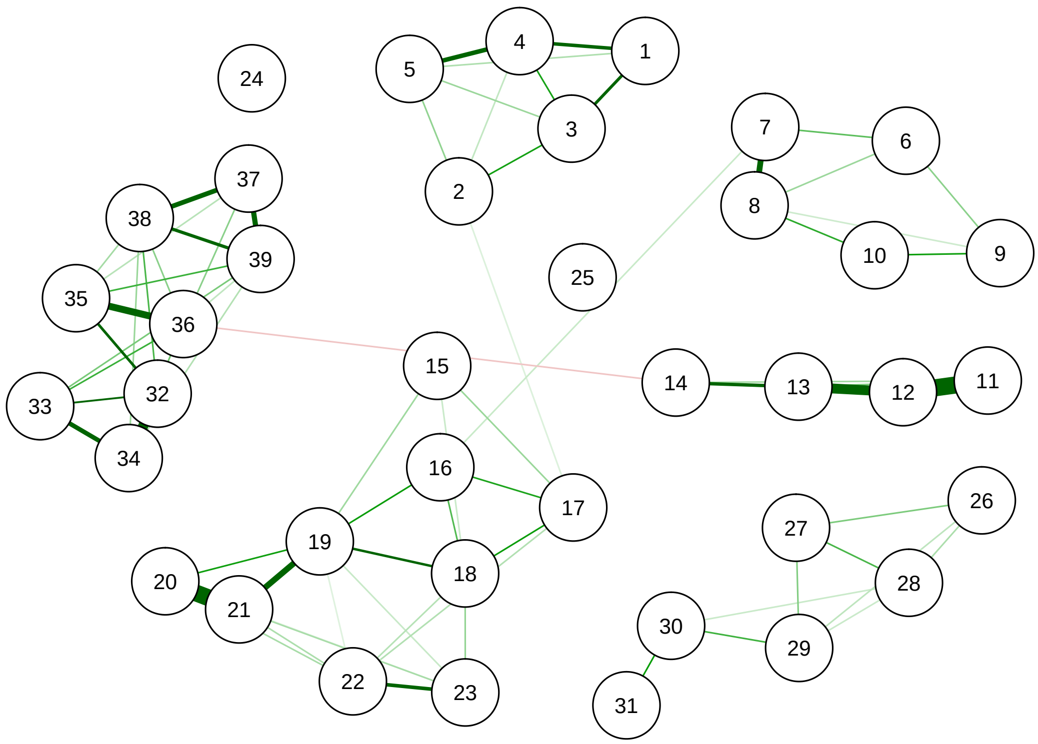

```
#################### Simulation input Col
Col_SimInput <- LinTransform(Col_fit$weiadj, Col_fit$thresholds)
#################### Connectivity simulation
set.seed(1)

Col_HighTemp <- IsingSampler(1000, Col_SimInput $ graph, rep(0,39), 1.8,
                             responses = c(-1L,1L))
Col_MidTemp <- IsingSampler(1000, Col_SimInput $ graph, rep(0,39), 1.2,
                            responses = c(-1L,1L))
Col_LowTemp <- IsingSampler(1000, Col_SimInput $ graph, rep(0,39), .6,
                            responses = c(-1L,1L))

# the simulated data contains -1;+1 responses. To estimate networks from the
# simulated data using IsingFit, 0s have to be assigned -1 responses
Col_HighTempResc <- Col_HighTemp
Col_HighTempResc[Col_HighTempResc == -1] <- 0

Col_MidTempResc <- Col_MidTemp
Col_MidTempResc[Col_MidTempResc == -1] <- 0

Col_LowTempResc <- Col_LowTemp
Col_LowTempResc[Col_LowTempResc == -1] <- 0

#these commands fit networks on the simulated data:
Col_HighTempFit <- IsingFit(Col_HighTempResc)
```




```
Col_MidTempFit <- IsingFit(Col_MidTempResc)
```




```
Col_LowTempFit <- IsingFit(Col_LowTempResc)
```




```r
# Figure S3: Connectivity Simulation
#tiff(filename = "Figure S3.tiff",
#     width = 8200, height = 6400, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
layout(matrix( c(1, 2, 3, 4,
                 5, 6, 7, 8,
                 9, 10, 11, 12), 3, 4, byrow = TRUE))

# LOW (Nl, then Col)
qgraph(NL_LowTempFit $ weiadj, layout = Nl_graph $ layout,
       maximum = max(abs(NL_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(NL_LowTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(NL_LowTemp, 1, sum), include.lowest =
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

```r
axis(1, seq(-30,35,5), seq(-30,35,5), cex.axis = .9)
axis(2)
mtext('Low Connectivity', line = 2, at = -17, font = 2)

qgraph(Col_LowTempFit $ weiadj, layout = Col_graph $ layout,
       maximum = max(abs(Col_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(Col_LowTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(Col_LowTemp, 1, sum), include.lowest =
```

```
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

```
axis(1, seq(-30,35,5), seq(-30,35,5), cex.axis = .9)
axis(2)
mtext('Low Connectivity', line = 2, at = -17, font = 2)

# MID (Nl, then Col)
qgraph(NL_MidTempFit $ weiadj, layout = Nl_graph $ layout,
       maximum = max(abs(NL_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(NL_MidTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(NL_MidTemp, 1, sum), include.lowest =
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

```
axis(1, seq(-30,35,5), seq(-30,35,5), cex.axis = .9)
axis(2)
mtext('Mid Connectivity', line = 2, at = -17, font = 2)

qgraph(Col_MidTempFit $ weiadj, layout = Col_graph $ layout,
       maximum = max(abs(Col_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(Col_MidTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(Col_MidTemp, 1, sum), include.lowest =
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

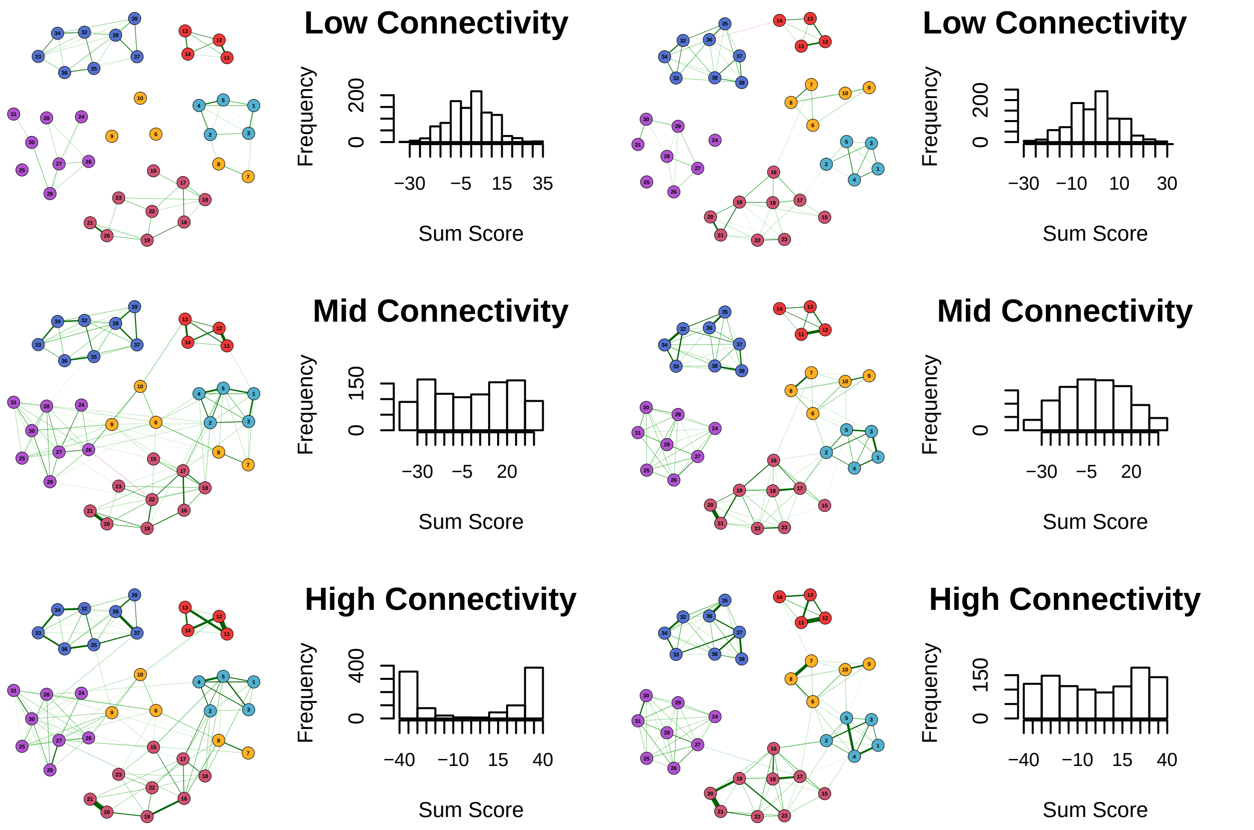
```
axis(1, seq(-30,35,5), seq(-30,35,5), cex.axis = .9)
axis(2)
mtext('Mid Connectivity', line = 2, at = -17, font = 2)


# HIGH (Nl, then Col)
qgraph(NL_HighTempFit $ weiadj, layout = Nl_graph $ layout,
       maximum = max(abs(NL_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(NL_HighTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(NL_HighTemp, 1, sum), include.lowest =
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

```
axis(1, seq(-40,40,5), seq(-40,40,5), cex.axis = .9)
axis(2)
mtext('High Connectivity', line = 2, at = -17, font = 2)

qgraph(Col_HighTempFit $ weiadj, layout = Col_graph $ layout,
       maximum = max(abs(Col_HighTempFit $ weiadj)),
       label.font = 2, color = group_col, groups = groups_type, legend = FALSE)
hist(apply(Col_HighTemp, 1, sum), include.lowest = FALSE, axes = FALSE,
     xlab = 'Sum Score', main = '')
```

```
## Warning in hist.default(apply(Col_HighTemp, 1, sum), include.lowest =
## FALSE, : 'include.lowest' ignored as 'breaks' is not a vector
```

```r
axis(1, seq(-40,40,5), seq(-40,40,5), cex.axis = .9)
axis(2)
mtext('High Connectivity', line = 2, at = -17, font = 2)
```

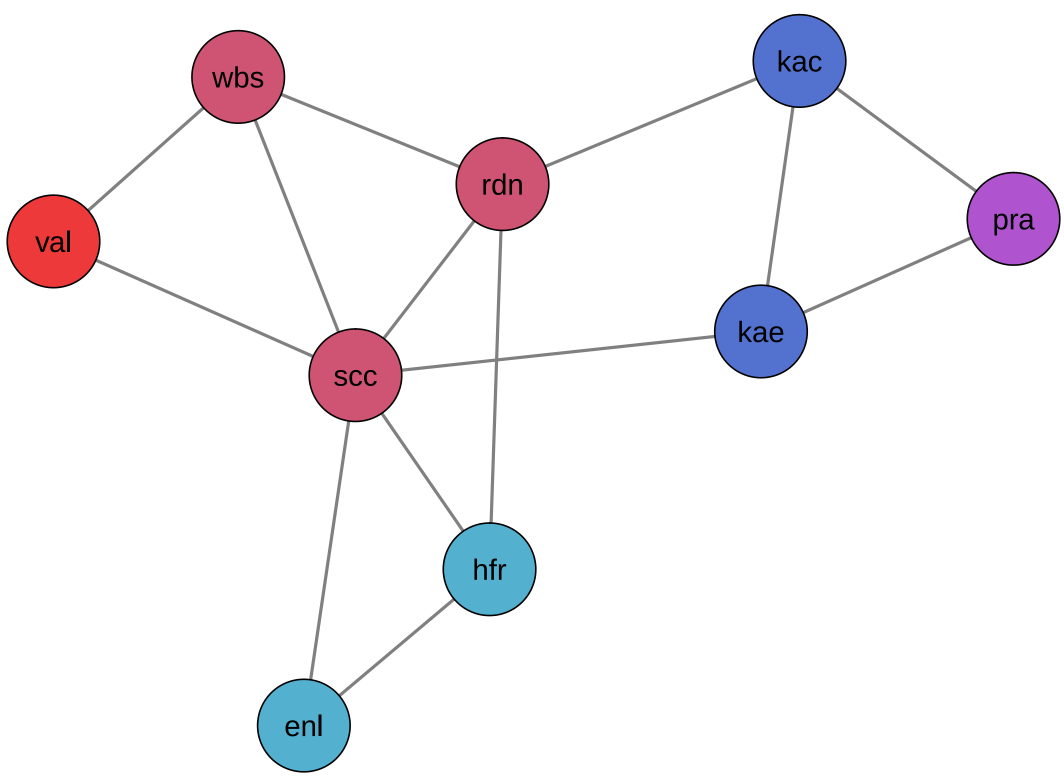



```r
#dev.off()
```

CAN Example: Figure 1

```r
data_paper <- matrix(c(0,1,1,1,1,1,1,0,0,
                       1,0,1,0,1,0,0,1,0,
                       1,1,0,0,0,1,0,0,0,
                       1,0,0,0,1,0,0,0,0,
                       1,1,0,1,0,0,0,0,0,
                       1,0,1,0,0,0,0,0,0,
                       1,0,0,0,0,0,0,1,1,
                       0,1,0,0,0,0,1,0,1,
                       0,0,0,0,0,0,1,1,0),9,9,byrow=TRUE)
colnames(data_paper) <- c("science center", "reading", "websites", "enjoy learning", "having fun reading
                       "value", "knowledge about effects", "knowledge about causes",
                        "perceived ability")
rownames(data_paper) <- c("science center", "reading", "websites", "enjoy learning", "having fun reading
                       "value", "knowledge about effects", "knowledge about causes",
                        "perceived ability")

groups_type <- list("G1"=c(1,2,3),
```

```r
                   "G2" = c(4,5),
                   "G3"=c(6),
                   "G4" = c(7,8),
                   "G5" = c(9))

# pick some nice colors
group_col <- c("#cf5372","#53B0CF",  "#ED3939","#5372cf","#b053cf")

names <- c("science center", "reading", "websites", "enjoy learning", "having fun reading",
                       "value", "knowledge about effects", "knowledge about causes",
                        "perceived ability")

# This command fits the network model:

#tiff(filename = "Figure 1.tiff",
#    width = 6400, height = 6400, units = "px", res = 800,
#     compression = c("none"),
#     bg = "white",
#     type = c("quartz"))
graph <- qgraph(data_paper,
                layout = 'spring',
                color = group_col,
                groups = groups_type,
                nodeNames = names,
                legend = FALSE,
                legend.mode="style2",
                legend.cex=.3)
```



```r
#dev.off()
```