EFA_Rcode_YZ_2020

Yicong Zhu

6/14/2020

Science and Mathematics Teacher Communities of Practice: Social Influences on Discipline-based Identity and Self-efficacy Beliefs

This is an R Markdown document for sharing the code applied in our Communities of Practice Paper.

We first need to load our Dataset (n = 165) and the required packages used in R Software.

```
library(psych)
library(MVN)
library(usdm)
library(Hmisc)
library(corrplot)
library(lavaan)
library(semPlot)
library(semTools)
library(dplyr)
library(ggpubr)
library(kableExtra)
library(igraph)
load(file = "~/20200614_EFA_Data.RData")
```

1. Demographic Summary

```
# Gender: 1 for female, 0 for male
(counts1 <- table(demographic$gender)) # 97 68
```

```
##
##  0  1
## 97 68
```

```
prop.table(counts1, margin=NULL)
```

```
##
##         0         1
## 0.5878788 0.4121212
```

```
# Race: white & non-Hispanic, Race 5 for white, ethnicity 1 for non-Hispanic
demographic$white <- ifelse(demographic$race==5 & demographic$ethnicity == 0, 1, 0)
(counts2 <- table(demographic$white)) # 10 155
```

```
##
##   0   1
##  10 155
```

```
prop.table(counts2, margin=NULL)
```

```
##
##          0          1
```

```
## 0.06060606 0.93939394
```

```r
# Did you receive a stipend/scholarship toward completing a teaching degree
# program and/or obtaining teacher certification/licensure?: 1 for yes, 0 for no
(counts3 <- table(demographic$ss)) # 66 99
```

```
##
##  0  1
## 66 99
```

```r
prop.table(counts3, margin=NULL)
```

```
##
##   0   1
## 0.4 0.6
```

```r
# Did you receive a salary supplement toward remaining in the
# teaching profession?: 1 for yes, 0 for no
(counts4 <- table(demographic$sup)) # 142 23
```

```
##
##   0   1
## 142  23
```

```r
prop.table(counts4, margin=NULL)
```

```
##
##         0         1
## 0.8606061 0.1393939
```

```r
# Field: 1 for science, 2 for math
(counts5 <- table(demographic$Math_or_Science)) # 36 129
```

```
##
##   1   2
##  36 129
```

```r
prop.table(counts5, margin=NULL)
```

```
##
##         1         2
## 0.2181818 0.7818182
```

```r
# Teaching in Full-time
# Current Teacher, Do you teach pre-K12 or not? 1 for yes, 0 for no
# Full-time Teacher, Are you teaching full time or part-time? 1 for full-time, 0 for part-time
table(demographic$teach) # 15 150
```

```
##
##   0   1
##  15 150
```

```r
demographic$fulltime[is.na(demographic$fulltime)] = 0
(counts6 <- table(demographic$fulltime)) # 24 141
```

```
##
##   0   1
##  24 141
```

```r
prop.table(counts6, margin=NULL)
```

```
##
##         0         1
## 0.1454545 0.8545455
```

```r
# Early Career Teacher, teaching year <= 5
demographic$early <- ifelse(demographic$year <= 5, 1, 0) # 1 for yes, 0 for no
(counts7 <- table(demographic$early)) # 22 143
```

```
##
##   0   1
##  22 143
```

```r
prop.table(counts7, margin=NULL)
```

```
##
##         0         1
## 0.1333333 0.8666667
```

```r
# Age
summary(demographic$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.00   27.00   31.00   33.84   39.00   63.00
```

2. EFA Details

2.1 Teacher Identity

```r
## Step 1: load Teacher Identity Data
STI_math <- data[,c(1,44:61)]
names(STI_math)[2:19] <- paste(c("x"), 1:18, sep="")
STI_sci <- data[,c(1,62:79)]
names(STI_sci)[2:19] <- paste(c("x"), 1:18, sep="")
STI = rbind(STI_math, STI_sci)
STI <- na.omit(STI)
Data_SMTI <- STI[,-1]
SMTI_id = STI$id

## Step 2: Pre-test before EFA
psych::describe(Data_SMTI)
```

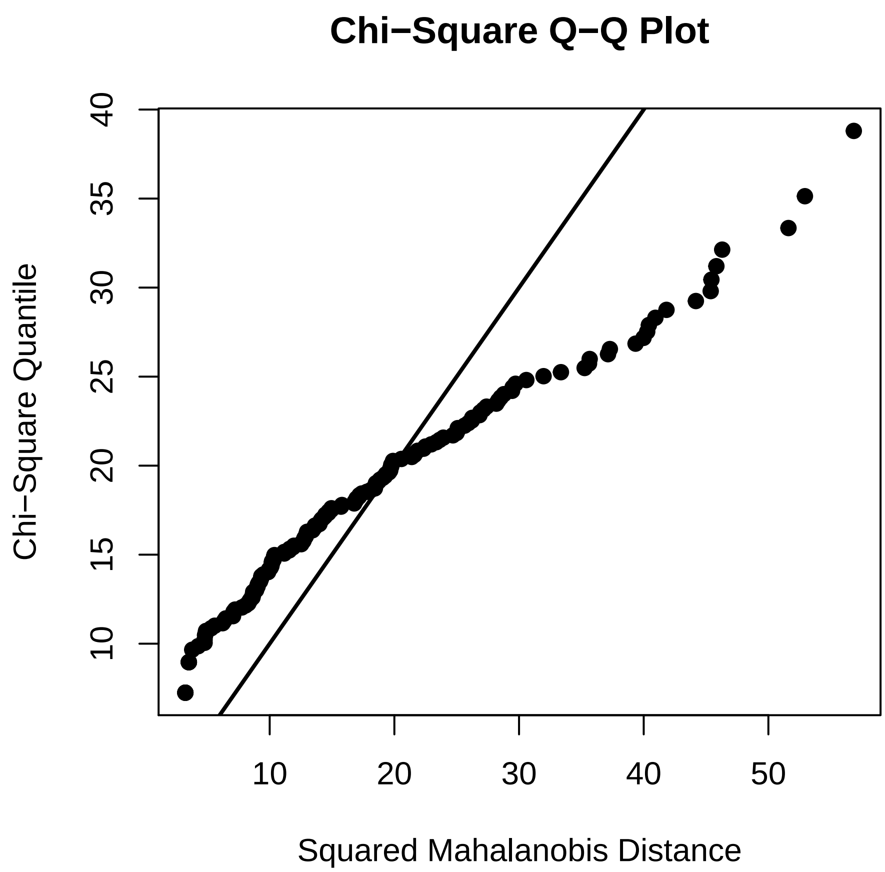
```
##     vars   n mean   sd median trimmed  mad min max range  skew kurtosis
## x1     1 165 3.61 0.55      4    3.66 0.00   1   4     3 -1.19     1.66
## x2     2 165 3.45 0.67      4    3.54 0.00   1   4     3 -1.04     0.89
## x3     3 165 3.24 0.84      3    3.34 1.48   1   4     3 -0.83    -0.15
## x4     4 165 2.90 1.05      3    3.00 1.48   1   4     3 -0.52    -0.99
## x5     5 165 3.50 0.74      4    3.65 0.00   1   4     3 -1.46     1.66
## x6     6 165 3.71 0.54      4    3.80 0.00   1   4     3 -1.91     3.94
## x7     7 165 3.73 0.50      4    3.81 0.00   1   4     3 -1.93     4.72
## x8     8 165 3.04 0.78      3    3.11 0.00   1   4     3 -0.62     0.18
## x9     9 165 3.47 0.62      4    3.54 0.00   1   4     3 -0.89     0.49
## x10   10 165 3.53 0.67      4    3.64 0.00   1   4     3 -1.47     2.23
```

```
## x11   11 165 3.26 0.71      3    3.34 1.48    1    4     3 -0.62    -0.07
## x12   12 165 3.47 0.67      4    3.57 0.00    1    4     3 -0.98     0.28
## x13   13 165 3.10 0.84      3    3.20 1.48    1    4     3 -0.75     0.04
## x14   14 165 3.45 0.69      4    3.56 0.00    1    4     3 -1.07     0.67
## x15   15 165 3.58 0.73      4    3.74 0.00    1    4     3 -1.74     2.50
## x16   16 165 3.61 0.71      4    3.77 0.00    1    4     3 -1.78     2.43
## x17   17 165 3.47 0.76      4    3.61 0.00    1    4     3 -1.33     1.11
## x18   18 165 3.57 0.65      4    3.68 0.00    1    4     3 -1.47     2.03
##       se
## x1  0.04
## x2  0.05
## x3  0.07
## x4  0.08
## x5  0.06
## x6  0.04
## x7  0.04
## x8  0.06
## x9  0.05
## x10 0.05
## x11 0.05
## x12 0.05
## x13 0.07
## x14 0.05
## x15 0.06
## x16 0.06
## x17 0.06
## x18 0.05
```

```
response.frequencies(Data_SMTI)
```

```
##               1           2         3         4 miss
## x1  0.006060606 0.012121212 0.3515152 0.6303030    0
## x2  0.012121212 0.060606061 0.3939394 0.5333333    0
## x3  0.036363636 0.151515152 0.3515152 0.4606061    0
## x4  0.139393939 0.187878788 0.3030303 0.3696970    0
## x5  0.024242424 0.072727273 0.2787879 0.6242424    0
## x6  0.006060606 0.024242424 0.2242424 0.7454545    0
## x7  0.006060606 0.006060606 0.2363636 0.7515152    0
## x8  0.042424242 0.151515152 0.5272727 0.2787879    0
## x9  0.006060606 0.048484848 0.4121212 0.5333333    0
## x10 0.018181818 0.042424242 0.3272727 0.6121212    0
## x11 0.012121212 0.115151515 0.4727273 0.4000000    0
## x12 0.006060606 0.078787879 0.3575758 0.5575758    0
## x13 0.054545455 0.139393939 0.4545455 0.3515152    0
## x14 0.012121212 0.078787879 0.3575758 0.5515152    0
## x15 0.024242424 0.066666667 0.2181818 0.6909091    0
## x16 0.018181818 0.078787879 0.1818182 0.7212121    0
## x17 0.024242424 0.090909091 0.2787879 0.6060606    0
## x18 0.012121212 0.048484848 0.2969697 0.6424242    0
```

```
# MVN: Mardia's test of skewness and kurtosis
mvn(Data_SMTI, multivariatePlot = "qq") # No
```




```
## $multivariateNormality
##              Test        Statistic               p value Result
## 1 Mardia Skewness 2558.42815571787 1.69091015004469e-110     NO
## 2 Mardia Kurtosis   23.62564816196                     0     NO
## 3             MVN             <NA>                  <NA>     NO
##
## $univariateNormality
##            Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk    x1        0.6465  <0.001      NO
## 2  Shapiro-Wilk    x2        0.7303  <0.001      NO
## 3  Shapiro-Wilk    x3        0.7936  <0.001      NO
## 4  Shapiro-Wilk    x4        0.8337  <0.001      NO
## 5  Shapiro-Wilk    x5        0.6837  <0.001      NO
## 6  Shapiro-Wilk    x6        0.5692  <0.001      NO
## 7  Shapiro-Wilk    x7        0.5494  <0.001      NO
## 8  Shapiro-Wilk    x8        0.8216  <0.001      NO
## 9  Shapiro-Wilk    x9        0.7221  <0.001      NO
## 10 Shapiro-Wilk    x10       0.6783  <0.001      NO
## 11 Shapiro-Wilk    x11       0.7920  <0.001      NO
## 12 Shapiro-Wilk    x12       0.7277  <0.001      NO
## 13 Shapiro-Wilk    x13       0.8187  <0.001      NO
## 14 Shapiro-Wilk    x14       0.7314  <0.001      NO
## 15 Shapiro-Wilk    x15       0.6278  <0.001      NO
## 16 Shapiro-Wilk    x16       0.6037  <0.001      NO
## 17 Shapiro-Wilk    x17       0.7023  <0.001      NO
## 18 Shapiro-Wilk    x18       0.6649  <0.001      NO
```

```
##
## $Descriptives
##       n     Mean   Std.Dev Median Min Max 25th 75th       Skew    Kurtosis
## x1  165 3.606061 0.5488010      4   1   4    3    4 -1.1930637  1.66095411
## x2  165 3.448485 0.6663155      4   1   4    3    4 -1.0417520  0.89478376
## x3  165 3.236364 0.8402301      3   1   4    3    4 -0.8304202 -0.14831260
## x4  165 2.903030 1.0547468      3   1   4    2    4 -0.5204971 -0.98930430
## x5  165 3.503030 0.7376978      4   1   4    3    4 -1.4596399  1.65526477
## x6  165 3.709091 0.5412067      4   1   4    3    4 -1.9121797  3.93898925
## x7  165 3.733333 0.4955083      4   1   4    4    4 -1.9257513  4.71599672
## x8  165 3.042424 0.7757885      3   1   4    3    4 -0.6175372  0.17915798
## x9  165 3.472727 0.6204211      4   1   4    3    4 -0.8871287  0.48575687
## x10 165 3.533333 0.6674792      4   1   4    3    4 -1.4660875  2.23381985
## x11 165 3.260606 0.7062178      3   1   4    3    4 -0.6171797 -0.06895993
## x12 165 3.466667 0.6674792      4   1   4    3    4 -0.9795052  0.27972727
## x13 165 3.103030 0.8382927      3   1   4    3    4 -0.7489910  0.03922028
## x14 165 3.448485 0.6932256      4   1   4    3    4 -1.0719453  0.67182099
## x15 165 3.575758 0.7254226      4   1   4    3    4 -1.7393982  2.50491965
## x16 165 3.606061 0.7130916      4   1   4    3    4 -1.7776378  2.43273026
## x17 165 3.466667 0.7613638      4   1   4    3    4 -1.3301293  1.10800963
## x18 165 3.569697 0.6460409      4   1   4    3    4 -1.4720744  2.02991789
```

```
# KMO
KMO(cor(Data_SMTI)) # Overall MSA =  0.87
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor(Data_SMTI))
## Overall MSA =  0.87
## MSA for each item =
##   x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11  x12  x13  x14  x15
## 0.86 0.88 0.87 0.90 0.77 0.88 0.81 0.90 0.89 0.79 0.58 0.90 0.91 0.91 0.88
##  x16  x17  x18
## 0.89 0.93 0.92
```

```
# Bartlett's test
cortest.bartlett(Data_SMTI)
```

```
## R was not square, finding R from data
```

```
## $chisq
## [1] 1619.733
##
## $p.value
## [1] 3.739935e-243
##
## $df
## [1] 153
```

```
# Multicolliearity
vif(Data_SMTI)
```

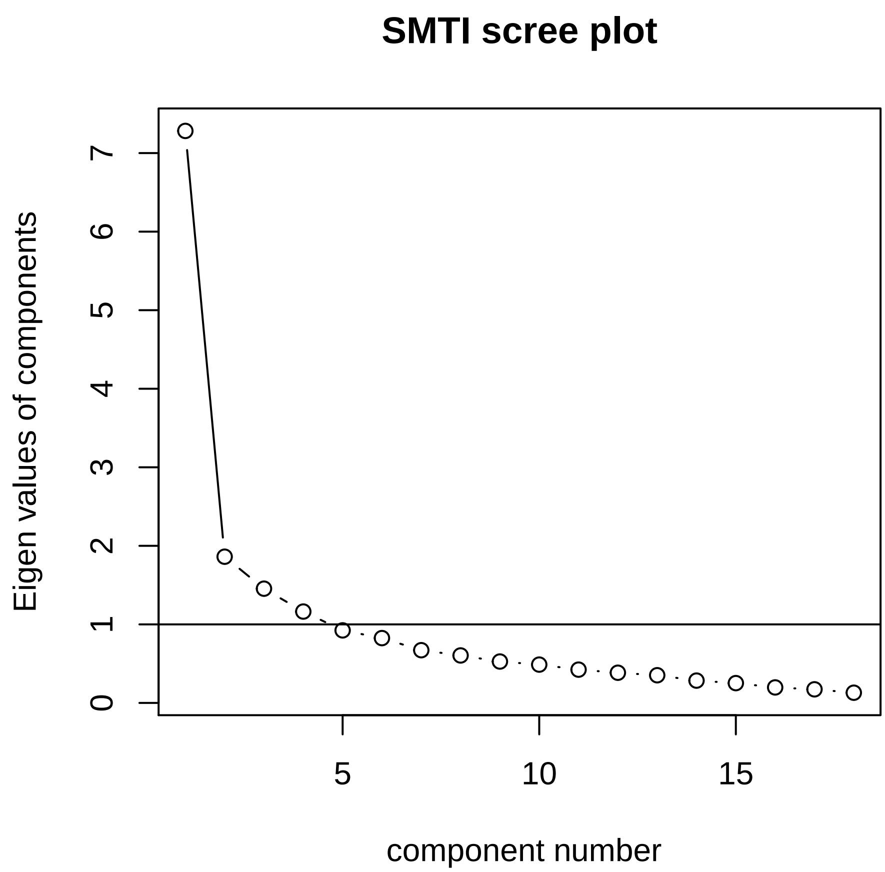
```
##    Variables      VIF
## 1         x1 1.876510
## 2         x2 2.016936
## 3         x3 2.867080
## 4         x4 1.620932
## 5         x5 2.525451
```

```
## 6          x6 1.621932
## 7          x7 1.928833
## 8          x8 1.785132
## 9          x9 2.356508
## 10        x10 1.926921
## 11        x11 2.044381
## 12        x12 2.166022
## 13        x13 1.538290
## 14        x14 2.619049
## 15        x15 5.334004
## 16        x16 4.181655
## 17        x17 4.099893
## 18        x18 2.906622
```

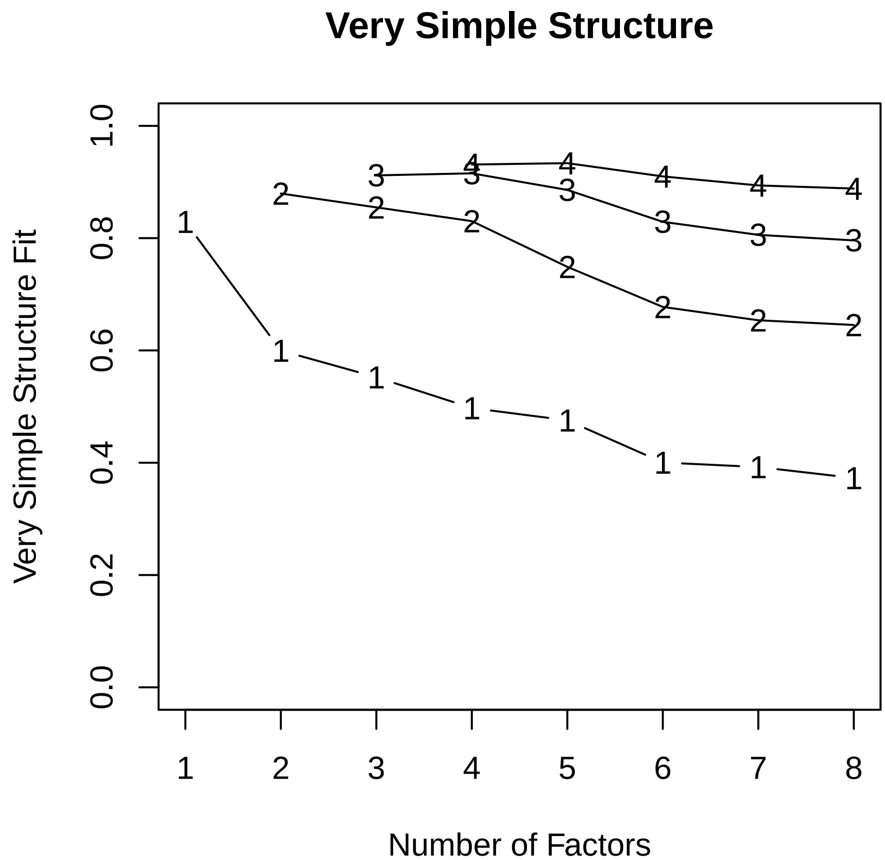
```
## Step 3: Determine the number of factors
# PCA: 4 factors
SMTI_pca <- prcomp(scale(Data_SMTI))
summary(SMTI_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.6986 1.3644 1.20607 1.07854 0.96108 0.90861
## Proportion of Variance 0.4046 0.1034 0.08081 0.06463 0.05132 0.04587
## Cumulative Proportion  0.4046 0.5080 0.58881 0.65343 0.70475 0.75061
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.8194 0.77752 0.72597 0.69852 0.65105 0.62009
## Proportion of Variance 0.0373 0.03359 0.02928 0.02711 0.02355 0.02136
## Cumulative Proportion  0.7879 0.82150 0.85078 0.87788 0.90143 0.92279
##                           PC13   PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.59373 0.5333 0.50227 0.44425 0.41613 0.36058
## Proportion of Variance 0.01958 0.0158 0.01402 0.01096 0.00962 0.00722
## Cumulative Proportion  0.94238 0.9582 0.97219 0.98316 0.99278 1.00000
```

```
# Scree Test: 4 factors
VSS.scree(Data_SMTI, main = "SMTI scree plot")
```



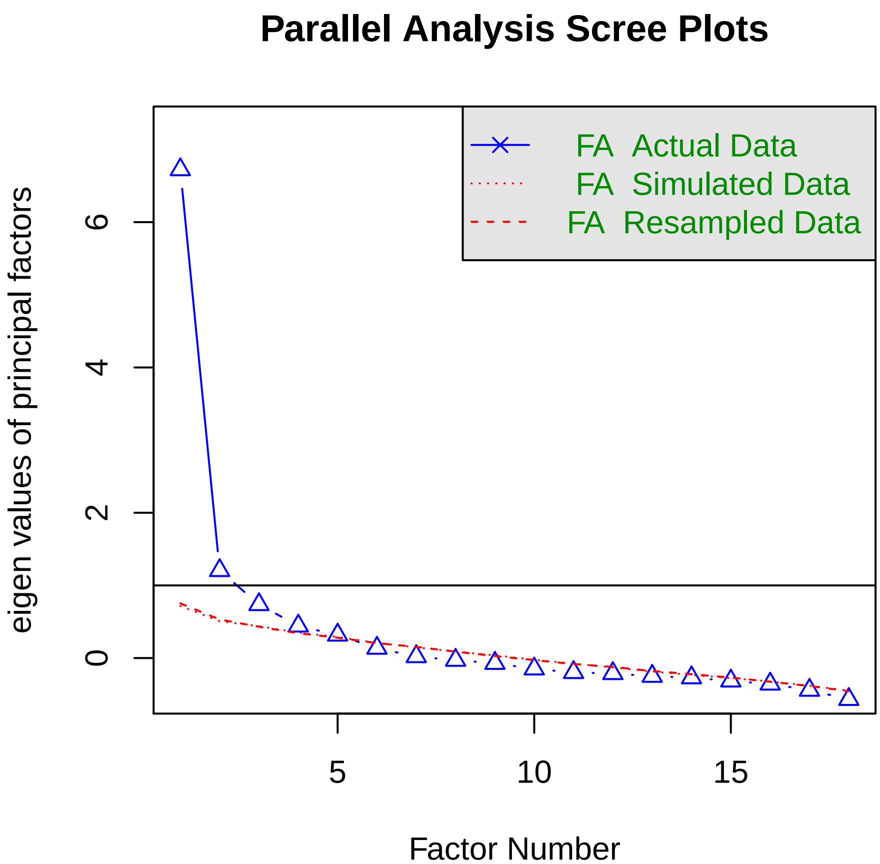

```r
# The Velicer MAP: 3 factors
# BIC: 3 factors
vss(Data_SMTI, fm = "pa")
```




```
##
## Very Simple Structure
## Call: vss(x = Data_SMTI, fm = "pa")
## VSS complexity 1 achieves a maximimum of 0.83  with  1  factors
## VSS complexity 2 achieves a maximimum of 0.88  with  2  factors
##
## The Velicer MAP achieves a minimum of 0.03  with  3  factors
## BIC achieves a minimum of  -258.61  with  3  factors
## Sample Size adjusted BIC achieves a minimum of  -37.16  with  6  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq    prob sqresid  fit RMSEA  BIC SABIC complex
## 1 0.83 0.00 0.037 135   592 2.0e-58    10.8 0.83 0.148  -98 329.8     1.0
## 2 0.60 0.88 0.031 118   371 7.2e-28     7.7 0.88 0.119 -232 141.7     1.4
## 3 0.55 0.85 0.030 102   262 4.6e-16     5.6 0.91 0.102 -259  64.3     1.6
## 4 0.50 0.83 0.033  87   206 1.1e-11     4.4 0.93 0.096 -238  37.4     1.7
## 5 0.48 0.75 0.037  73   143 2.0e-06     3.6 0.94 0.081 -230   1.1     1.9
## 6 0.40 0.68 0.041  60    79 4.9e-02     3.1 0.95 0.050 -227 -37.2     2.1
## 7 0.39 0.65 0.052  48    61 9.9e-02     2.6 0.96 0.047 -184 -32.2     2.1
## 8 0.37 0.65 0.062  37    39 3.7e-01     2.4 0.96 0.030 -150 -32.6     2.2
##   eChisq  SRMR eCRMS eBIC
## 1  545.4 0.104 0.111 -144
## 2  277.6 0.074 0.084 -325
## 3  146.5 0.054 0.066 -374
## 4   85.7 0.041 0.055 -359
## 5   49.0 0.031 0.045 -324
```

```
## 6   22.3 0.021 0.034 -284
## 7   14.5 0.017 0.030 -231
## 8    8.7 0.013 0.027 -180
```

```
# Parallel analysis: 4 or 5 factors
parallel_SMTI = fa.parallel(Data_SMTI,fm = "pa", fa = "fa")
```




```
## Parallel analysis suggests that the number of factors =  4  and the number of components =  NA
```

```
## Step 4: SMTI EFA procedure
# fa_SMTI_1 = fa(Data_SMTI, nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_1, digits = 3, cut = 0.45) # 8,9,12
#
# fa_SMTI_2 = fa(Data_SMTI[,-c(12)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_2, digits = 3, cut = 0.45) # 8,9
#
# fa_SMTI_3 = fa(Data_SMTI[,-c(12,8)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_3, digits = 3, cut = 0.45) # 9
#
# fa_SMTI_4 = fa(Data_SMTI[,-c(12,8,9)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_4, digits = 3, cut = 0.45) # 2
#
# fa_SMTI_5 = fa(Data_SMTI[,-c(12,8,9,2)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_5, digits = 3, cut = 0.45) # 1
#
# fa_SMTI_6 = fa(Data_SMTI[,-c(12,8,9,2,1)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_SMTI_6, digits = 3, cut = 0.45)
#
```

```r
fa_SMTI_FF = fa(Data_SMTI[,-c(12,8,9,2,1)], nfactors = 4, fm = "pa", rotate = "oblimin")
```

```
## maximum iteration exceeded
## Loading required namespace: GPArotation
```

```r
print(fa_SMTI_FF, digits = 3, cut=0.45)
```

```
## Factor Analysis using method =  pa
## Call: fa(r = Data_SMTI[, -c(12, 8, 9, 2, 1)], nfactors = 4, rotate = "oblimin",
##     fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##        PA1    PA2    PA3    PA4    h2     u2  com
## x3           0.792                0.765 0.2350 1.09
## x4           0.488                0.345 0.6550 1.18
## x5           0.804                0.655 0.3447 1.09
## x6                         0.470 0.385 0.6151 1.35
## x7                         0.953 0.937 0.0629 1.00
## x10                 0.729        0.604 0.3955 1.11
## x11                 0.806        0.660 0.3401 1.05
## x13  0.484                       0.318 0.6820 1.65
## x14  0.696                       0.580 0.4202 1.22
## x15  0.900                       0.859 0.1407 1.06
## x16  0.818                       0.735 0.2650 1.04
## x17  0.882                       0.778 0.2218 1.01
## x18  0.754                       0.615 0.3855 1.14
##
##                         PA1   PA2   PA3   PA4
## SS loadings           3.799 1.724 1.400 1.314
## Proportion Var        0.292 0.133 0.108 0.101
## Cumulative Var        0.292 0.425 0.532 0.634
## Proportion Explained  0.461 0.209 0.170 0.160
## Cumulative Proportion 0.461 0.670 0.840 1.000
##
##  With factor correlations of
##       PA1   PA2   PA3   PA4
## PA1 1.000 0.449 0.155 0.326
## PA2 0.449 1.000 0.271 0.436
## PA3 0.155 0.271 1.000 0.225
## PA4 0.326 0.436 0.225 1.000
##
## Mean item complexity =  1.2
## Test of the hypothesis that 4 factors are sufficient.
##
## The degrees of freedom for the null model are  78  and the objective function was  7.183 with Chi Sq
## The degrees of freedom for the model are 32  and the objective function was  0.402
##
## The root mean square of the residuals (RMSR) is  0.027
## The df corrected root mean square of the residuals is  0.042
##
## The harmonic number of observations is  165 with the empirical chi square  18.679  with prob <  0.97
## The total number of observations was  165  with Likelihood Chi Square =  62.844  with prob <  0.0009
##
## Tucker Lewis Index of factoring reliability =  0.928
## RMSEA index =  0.0805  and the 90 % confidence intervals are  0.0481 0.1045
```

```
## BIC =  -100.546
## Fit based upon off diagonal values = 0.995
## Measures of factor score adequacy
##                                                    PA1   PA2   PA3   PA4
## Correlation of (regression) scores with factors   0.969 0.921 0.887 0.970
## Multiple R square of scores with factors          0.938 0.849 0.787 0.941
## Minimum correlation of possible factor scores     0.876 0.697 0.574 0.881
```

```
## Step 5: Summed up items scores
PA1 = c("x13","x14","x15","x16","x17","x18")
PA2 = c("x3","x4","x5")
PA3 = c("x10","x11")
PA4 = c("x6","x7")
# PA1: SI, Self Image
# PA2: CA, Community of Practice 1, Community Action
# PA3: SR, Social Respect
# PA4: CV, Community of Practice 2, Community Value
SI <- rowSums(Data_SMTI[PA1])
CA <- rowSums(Data_SMTI[PA2])
SR <- rowSums(Data_SMTI[PA3])
CV <- rowSums(Data_SMTI[PA4])
SMTI <- rowSums(Data_SMTI[c(PA1, PA2, PA3, PA4)])
SMTI_13 = data.frame(SMTI_id,SI,CA,SR,CV,SMTI)
names(SMTI_13)[1] <- "id"
data <- merge(data, SMTI_13, by = "id")
psych::describe(data[c("SI","CA","SR","CV","SMTI")])
```

```
##      vars   n  mean   sd median trimmed  mad min max range  skew kurtosis
## SI      1 165 20.77 3.55     22   21.37 2.97   6  24    18 -1.54     2.55
## CA      2 165  9.64 2.19     10    9.92 2.97   3  12     9 -0.93     0.40
## SR      3 165  6.79 1.23      7    6.95 1.48   2   8     6 -1.12     1.68
## CV      4 165  7.44 0.92      8    7.61 0.00   2   8     6 -2.10     6.54
## SMTI    5 165 44.65 5.97     46   45.32 5.93  13  52    39 -1.42     3.62
##        se
## SI   0.28
## CA   0.17
## SR   0.10
## CV   0.07
## SMTI 0.46
```

```
# mvn(data[c("SI","CA","SR","CV","SMTI")], multivariatePlot = "qq")

## Step 6: Reliability Test
alpha.pa1 = psych::alpha(Data_SMTI[PA1]) # SI
print(alpha.pa1$total, digits = 3) # 0.894
```

```
##  raw_alpha std.alpha G6(smc) average_r S/N    ase mean    sd median_r
##      0.894     0.899   0.903     0.597 8.9 0.0131 3.46 0.592    0.609
```

```
alpha.pa2 = psych::alpha(Data_SMTI[PA2]) # CA
print(alpha.pa2$total, digits = 3) # 0.760
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N    ase mean   sd median_r
##       0.76     0.781    0.72     0.542 3.56 0.0324 3.21 0.73    0.508
```

```
alpha.pa3 = psych::alpha(Data_SMTI[PA3]) # SR
print(alpha.pa3$total, digits = 3) # 0.756
```

```
## raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##     0.756     0.757   0.609     0.609 3.11 0.0379  3.4 0.616    0.609
```

```r
alpha.pa4 = psych::alpha(Data_SMTI[PA4]) # CV
print(alpha.pa4$total, digits = 3) # 0.727
```

```
## raw_alpha std.alpha G6(smc) average_r  S/N    ase mean   sd median_r
##     0.727     0.729   0.573     0.573 2.68 0.0423 3.72 0.46    0.573
```

```r
alpha.patial_all = psych::alpha(Data_SMTI[c(PA1,PA2,PA3,PA4)]) # all 13 items
print(alpha.patial_all$total, digits = 3) # 0.869
```

```
## raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##     0.869     0.874   0.912     0.348 6.94 0.0148 3.43 0.459     0.31
```

```r
## Step 7: Special Reliability Test for SMTI: SR COP2
# Peasron correlation
cor.test(as.matrix(Data_SMTI["x10"]),as.matrix(Data_SMTI["x11"])) # 0.6088261 <0.001
```

```
##
## 	Pearson's product-moment correlation
##
## data:  as.matrix(Data_SMTI["x10"]) and as.matrix(Data_SMTI["x11"])
## t = 9.7982, df = 163, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5028135 0.6967947
## sample estimates:
##       cor
## 0.6088261
```

```r
cor.test(as.matrix(Data_SMTI["x6"]),as.matrix(Data_SMTI["x7"])) # 0.5729842 <0.001
```

```
##
## 	Pearson's product-moment correlation
##
## data:  as.matrix(Data_SMTI["x6"]) and as.matrix(Data_SMTI["x7"])
## t = 8.9259, df = 163, p-value = 8.831e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4605153 0.6673465
## sample estimates:
##       cor
## 0.5729842
```

```r
# Spearman Brown Coefficient:
# https://www.r-bloggers.com/five-ways-to-calculate-internal-consistency/
score_e <- as.matrix(Data_SMTI[PA3])[, c(TRUE, FALSE)]  # with even items
score_o <- as.matrix(Data_SMTI[PA3])[, c(FALSE, TRUE)]  # with odd items
r1 <- cor(score_e, score_o)
spearman_brown_PA3 = (2 * r1) / (1 + r1)
spearman_brown_PA3 # 0.7568575
```

```
## [1] 0.7568575
```

```r
score_e <- as.matrix(Data_SMTI[PA4])[, c(TRUE, FALSE)]  # with even items
score_o <- as.matrix(Data_SMTI[PA4])[, c(FALSE, TRUE)]  # with odd items
```

```r
r2 <- cor(score_e, score_o)
spearman_brown_PA4 = (2 * r2) / (1 + r2)
spearman_brown_PA4 # 0.7285314
```

```
## [1] 0.7285314
```

2.2 Teacher Self-Efficacy

```r
## Step 1: load Teacher Self-Efficacy Data
TEBS <- data[,c(1,13:43)]
names(TEBS)[2:32] <- paste(c("x"), 1:31, sep="")
Data_TEBS <- TEBS[,-1]

## Step 2: Pre-test before EFA
psych::describe(Data_TEBS)
```

```
##     vars   n mean   sd median trimmed  mad min max range  skew kurtosis
## x1     1 165 2.59 0.77      3    2.56 1.48   1   4     3  0.20    -0.54
## x2     2 165 2.54 0.79      2    2.50 1.48   1   4     3  0.35    -0.54
## x3     3 165 2.93 0.76      3    2.94 1.48   1   4     3 -0.21    -0.54
## x4     4 165 2.99 0.79      3    3.03 1.48   1   4     3 -0.42    -0.30
## x5     5 165 3.10 0.73      3    3.13 1.48   1   4     3 -0.24    -0.80
## x6     6 165 2.75 0.83      3    2.75 1.48   1   4     3 -0.07    -0.71
## x7     7 165 2.60 0.85      3    2.61 1.48   1   4     3  0.04    -0.70
## x8     8 165 3.06 0.90      3    3.14 1.48   1   4     3 -0.51    -0.78
## x9     9 165 3.20 0.73      3    3.27 1.48   1   4     3 -0.60     0.02
## x10   10 165 3.01 0.81      3    3.03 1.48   1   4     3 -0.22    -0.96
## x11   11 165 2.95 0.85      3    2.98 1.48   1   4     3 -0.21    -0.97
## x12   12 165 2.79 0.81      3    2.79 1.48   1   4     3 -0.08    -0.69
## x13   13 165 2.84 0.77      3    2.82 1.48   1   4     3  0.04    -0.84
## x14   14 165 2.73 0.83      3    2.74 1.48   1   4     3 -0.04    -0.74
## x15   15 165 3.39 0.72      4    3.51 0.00   1   4     3 -1.03     0.67
## x16   16 165 3.06 0.72      3    3.09 0.00   1   4     3 -0.28    -0.49
## x17   17 165 2.78 0.83      3    2.79 1.48   1   4     3 -0.14    -0.66
## x18   18 165 2.87 0.81      3    2.89 1.48   1   4     3 -0.26    -0.53
## x19   19 165 2.67 0.86      3    2.68 1.48   1   4     3 -0.04    -0.75
## x20   20 165 2.87 0.84      3    2.89 1.48   1   4     3 -0.18    -0.78
## x21   21 165 2.86 0.83      3    2.87 1.48   1   4     3 -0.11    -0.87
## x22   22 165 2.90 0.79      3    2.92 1.48   1   4     3 -0.25    -0.53
## x23   23 165 2.96 0.70      3    2.98 0.00   1   4     3 -0.27    -0.10
## x24   24 165 2.57 0.89      3    2.59 1.48   1   4     3  0.02    -0.79
## x25   25 165 2.77 0.79      3    2.77 1.48   1   4     3 -0.15    -0.51
## x26   26 165 2.67 0.80      3    2.67 1.48   1   4     3 -0.07    -0.52
## x27   27 165 2.65 0.79      3    2.62 1.48   1   4     3  0.13    -0.64
## x28   28 165 2.60 0.77      3    2.56 1.48   1   4     3  0.18    -0.54
## x29   29 165 3.12 0.69      3    3.17 0.00   1   4     3 -0.38    -0.10
## x30   30 165 3.07 0.79      3    3.14 0.00   1   4     3 -0.65     0.16
## x31   31 165 3.18 0.80      3    3.26 1.48   1   4     3 -0.69    -0.15
##       se
## x1  0.06
## x2  0.06
## x3  0.06
## x4  0.06
```

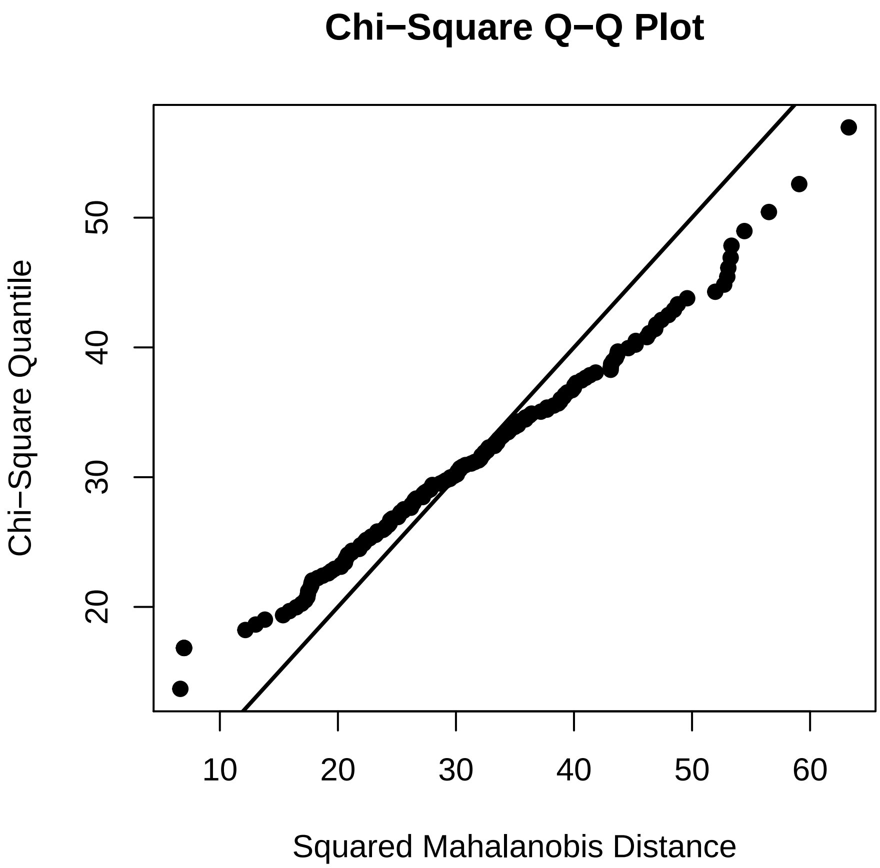
```
## x5  0.06
## x6  0.06
## x7  0.07
## x8  0.07
## x9  0.06
## x10 0.06
## x11 0.07
## x12 0.06
## x13 0.06
## x14 0.06
## x15 0.06
## x16 0.06
## x17 0.06
## x18 0.06
## x19 0.07
## x20 0.07
## x21 0.06
## x22 0.06
## x23 0.05
## x24 0.07
## x25 0.06
## x26 0.06
## x27 0.06
## x28 0.06
## x29 0.05
## x30 0.06
## x31 0.06
```

```
response.frequencies(Data_TEBS)
```

```
##               1          2         3         4 miss
## x1  0.048484848 0.43636364 0.3878788 0.1272727    0
## x2  0.054545455 0.48484848 0.3272727 0.1333333    0
## x3  0.024242424 0.25454545 0.4909091 0.2303030    0
## x4  0.036363636 0.20606061 0.4909091 0.2666667    0
## x5  0.006060606 0.20000000 0.4848485 0.3090909    0
## x6  0.054545455 0.33939394 0.4121212 0.1939394    0
## x7  0.084848485 0.38787879 0.3696970 0.1575758    0
## x8  0.048484848 0.23030303 0.3333333 0.3878788    0
## x9  0.018181818 0.12727273 0.4909091 0.3636364    0
## x10 0.018181818 0.26666667 0.4060606 0.3090909    0
## x11 0.030303030 0.29090909 0.3757576 0.3030303    0
## x12 0.042424242 0.32727273 0.4303030 0.2000000    0
## x13 0.018181818 0.33333333 0.4424242 0.2060606    0
## x14 0.054545455 0.35151515 0.4000000 0.1939394    0
## x15 0.018181818 0.08484848 0.3818182 0.5151515    0
## x16 0.012121212 0.19393939 0.5151515 0.2787879    0
## x17 0.054545455 0.31515152 0.4303030 0.2000000    0
## x18 0.042424242 0.26666667 0.4666667 0.2242424    0
## x19 0.078787879 0.35757576 0.3818182 0.1818182    0
## x20 0.042424242 0.29696970 0.4121212 0.2484848    0
## x21 0.036363636 0.31515152 0.4000000 0.2484848    0
## x22 0.036363636 0.26060606 0.4727273 0.2303030    0
## x23 0.018181818 0.20606061 0.5696970 0.2060606    0
## x24 0.109090909 0.37575758 0.3515152 0.1636364    0
```

```
## x25 0.048484848 0.30909091 0.4666667 0.1757576    0
## x26 0.060606061 0.35151515 0.4424242 0.1454545    0
## x27 0.048484848 0.40606061 0.3939394 0.1515152    0
## x28 0.048484848 0.43030303 0.3939394 0.1272727    0
## x29 0.012121212 0.14545455 0.5515152 0.2909091    0
## x30 0.042424242 0.14545455 0.5090909 0.3030303    0
## x31 0.030303030 0.15151515 0.4242424 0.3939394    0
```

```r
# MVN: Mardia's test of skewness and kurtosis
mvn(Data_TEBS, multivariatePlot = "qq") # No
```




```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 6814.60146258235 7.93169081464944e-34     NO
## 2 Mardia Kurtosis 9.40887519296675                    0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##            Test  Variable Statistic   p value Normality
## 1  Shapiro-Wilk    x1        0.8451  <0.001      NO
## 2  Shapiro-Wilk    x2        0.8368  <0.001      NO
## 3  Shapiro-Wilk    x3        0.8422  <0.001      NO
## 4  Shapiro-Wilk    x4        0.8408  <0.001      NO
## 5  Shapiro-Wilk    x5        0.8182  <0.001      NO
## 6  Shapiro-Wilk    x6        0.8633  <0.001      NO
## 7  Shapiro-Wilk    x7        0.8701  <0.001      NO
## 8  Shapiro-Wilk    x8        0.8308  <0.001      NO
```

```
## 9  Shapiro-Wilk    x9        0.8035  <0.001      NO
## 10 Shapiro-Wilk    x10       0.8366  <0.001      NO
## 11 Shapiro-Wilk    x11       0.8450  <0.001      NO
## 12 Shapiro-Wilk    x12       0.8577  <0.001      NO
## 13 Shapiro-Wilk    x13       0.8390  <0.001      NO
## 14 Shapiro-Wilk    x14       0.8630  <0.001      NO
## 15 Shapiro-Wilk    x15       0.7496  <0.001      NO
## 16 Shapiro-Wilk    x16       0.8231  <0.001      NO
## 17 Shapiro-Wilk    x17       0.8632  <0.001      NO
## 18 Shapiro-Wilk    x18       0.8544  <0.001      NO
## 19 Shapiro-Wilk    x19       0.8711  <0.001      NO
## 20 Shapiro-Wilk    x20       0.8574  <0.001      NO
## 21 Shapiro-Wilk    x21       0.8540  <0.001      NO
## 22 Shapiro-Wilk    x22       0.8509  <0.001      NO
## 23 Shapiro-Wilk    x23       0.8186  <0.001      NO
## 24 Shapiro-Wilk    x24       0.8762  <0.001      NO
## 25 Shapiro-Wilk    x25       0.8570  <0.001      NO
## 26 Shapiro-Wilk    x26       0.8605  <0.001      NO
## 27 Shapiro-Wilk    x27       0.8525  <0.001      NO
## 28 Shapiro-Wilk    x28       0.8461  <0.001      NO
## 29 Shapiro-Wilk    x29       0.8057  <0.001      NO
## 30 Shapiro-Wilk    x30       0.8204  <0.001      NO
## 31 Shapiro-Wilk    x31       0.8124  <0.001      NO
##
## $Descriptives
##       n     Mean   Std.Dev Median Min Max 25th 75th        Skew
## x1  165 2.593939 0.7722076      3   1   4    2    3  0.19977060
## x2  165 2.539394 0.7924720      2   1   4    2    3  0.34636837
## x3  165 2.927273 0.7616064      3   1   4    2    3 -0.20827569
## x4  165 2.987879 0.7885451      3   1   4    3    4 -0.42386042
## x5  165 3.096970 0.7260337      3   1   4    3    4 -0.24227669
## x6  165 2.745455 0.8312094      3   1   4    2    3 -0.07125250
## x7  165 2.600000 0.8539721      3   1   4    2    3  0.03970500
## x8  165 3.060606 0.9018745      3   1   4    2    4 -0.51463252
## x9  165 3.200000 0.7258301      3   1   4    3    4 -0.60481086
## x10 165 3.006061 0.8077141      3   1   4    2    4 -0.21789457
## x11 165 2.951515 0.8468454      3   1   4    2    4 -0.20830941
## x12 165 2.787879 0.8099072      3   1   4    2    3 -0.07948948
## x13 165 2.836364 0.7674070      3   1   4    2    3  0.04205295
## x14 165 2.733333 0.8347143      3   1   4    2    3 -0.03603293
## x15 165 3.393939 0.7215918      4   1   4    3    4 -1.03245487
## x16 165 3.060606 0.7215918      3   1   4    3    4 -0.28329410
## x17 165 2.775758 0.8289835      3   1   4    2    3 -0.14171551
## x18 165 2.872727 0.8051938      3   1   4    2    3 -0.25613128
## x19 165 2.666667 0.8648511      3   1   4    2    3 -0.03955789
## x20 165 2.866667 0.8376312      3   1   4    2    3 -0.18131398
## x21 165 2.860606 0.8329859      3   1   4    2    3 -0.11499334
## x22 165 2.896970 0.7934506      3   1   4    2    3 -0.25365007
## x23 165 2.963636 0.6974772      3   1   4    3    3 -0.27308227
## x24 165 2.569697 0.8918619      3   1   4    2    3  0.02173014
## x25 165 2.769697 0.7934506      3   1   4    2    3 -0.15346833
## x26 165 2.672727 0.7975850      3   1   4    2    3 -0.06911103
## x27 165 2.648485 0.7948467      3   1   4    2    3  0.12588223
## x28 165 2.600000 0.7714415      3   1   4    2    3  0.18058979
```

```
## x29 165 3.121212 0.6877666      3   1   4    3    4 -0.38195723
## x30 165 3.072727 0.7852577      3   1   4    3    4 -0.65244986
## x31 165 3.181818 0.7984186      3   1   4    3    4 -0.69083739
##        Kurtosis
## x1  -0.53848547
## x2  -0.54330861
## x3  -0.54010096
## x4  -0.30000628
## x5  -0.79609853
## x6  -0.70982227
## x7  -0.70390116
## x8  -0.78164283
## x9   0.02491957
## x10 -0.95759127
## x11 -0.96967998
## x12 -0.69402998
## x13 -0.84330319
## x14 -0.74379064
## x15  0.67435761
## x16 -0.48803542
## x17 -0.65627544
## x18 -0.52882959
## x19 -0.75154093
## x20 -0.78030696
## x21 -0.86558010
## x22 -0.52607549
## x23 -0.10250071
## x24 -0.78916328
## x25 -0.50717023
## x26 -0.52158742
## x27 -0.64147626
## x28 -0.53640061
## x29 -0.09994578
## x30  0.15517646
## x31 -0.14729589
```

```
# KMO
KMO(cor(Data_TEBS)) # Overall MSA =  0.94
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor(Data_TEBS))
## Overall MSA =  0.94
## MSA for each item =
##   x1   x2   x3   x4   x5   x6   x7   x8   x9  x10  x11  x12  x13  x14  x15
## 0.95 0.96 0.94 0.94 0.93 0.94 0.94 0.92 0.94 0.94 0.94 0.93 0.95 0.94 0.95
##  x16  x17  x18  x19  x20  x21  x22  x23  x24  x25  x26  x27  x28  x29  x30
## 0.95 0.91 0.93 0.97 0.96 0.93 0.94 0.97 0.95 0.95 0.94 0.87 0.90 0.95 0.94
##  x31
## 0.93
```

```
# Bartlett's test
cortest.bartlett(Data_TEBS)
```

```
## R was not square, finding R from data
```

```
## $chisq
```

```
## [1] 3778.872
##
## $p.value
## [1] 0
##
## $df
## [1] 465
```

```
# Multicolliearity
vif(Data_TEBS)
```

```
##    Variables      VIF
## 1         x1 2.752010
## 2         x2 2.327361
## 3         x3 2.304239
## 4         x4 3.125537
## 5         x5 2.969655
## 6         x6 3.009584
## 7         x7 3.159704
## 8         x8 3.597569
## 9         x9 3.440041
## 10       x10 2.977972
## 11       x11 2.651563
## 12       x12 2.961985
## 13       x13 2.447555
## 14       x14 3.221861
## 15       x15 1.929547
## 16       x16 2.558715
## 17       x17 3.830790
## 18       x18 3.333135
## 19       x19 3.452248
## 20       x20 2.962007
## 21       x21 3.468679
## 22       x22 2.293661
## 23       x23 3.135178
## 24       x24 3.196049
## 25       x25 3.961454
## 26       x26 3.492747
## 27       x27 3.245181
## 28       x28 3.823947
## 29       x29 2.899304
## 30       x30 3.564673
## 31       x31 3.406802
```

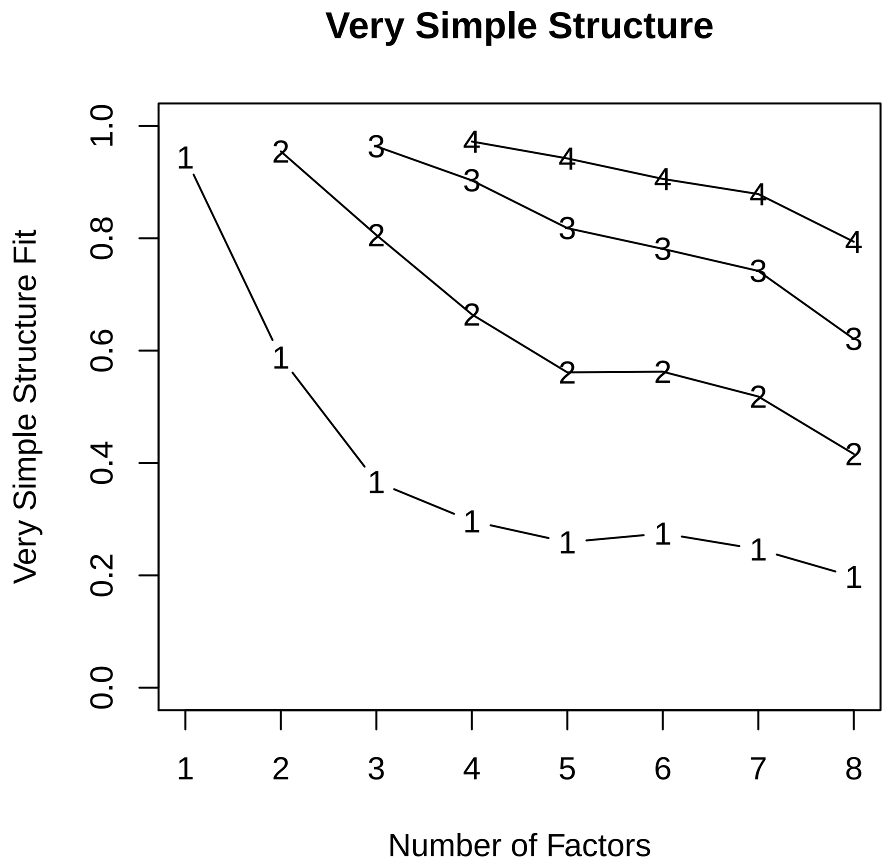
```
## Step 3: Determine the number of factors
# PCA: 5 factors
TEBS_pca <- prcomp(scale(Data_TEBS))
summary(TEBS_pca)
```

```
## Importance of components:
##                           PC1     PC2    PC3     PC4     PC5     PC6
## Standard deviation     3.9032 1.28687 1.2350 1.21647 1.02769 0.93721
## Proportion of Variance 0.4915 0.05342 0.0492 0.04774 0.03407 0.02833
## Cumulative Proportion  0.4915 0.54488 0.5941 0.64182 0.67588 0.70422
##                            PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.88011 0.84816 0.84337 0.79833 0.76501 0.71677
```

```
## Proportion of Variance 0.02499 0.02321 0.02294 0.02056 0.01888 0.01657
## Cumulative Proportion  0.72921 0.75241 0.77536 0.79591 0.81479 0.83137
##                           PC13    PC14   PC15    PC16    PC17    PC18
## Standard deviation     0.6865 0.66737 0.6421 0.62665 0.61129 0.59008
## Proportion of Variance 0.0152 0.01437 0.0133 0.01267 0.01205 0.01123
## Cumulative Proportion  0.8466 0.86094 0.8742 0.88690 0.89896 0.91019
##                           PC19    PC20    PC21   PC22    PC23    PC24
## Standard deviation     0.57978 0.55719 0.53628 0.5223 0.48910 0.45825
## Proportion of Variance 0.01084 0.01001 0.00928 0.0088 0.00772 0.00677
## Cumulative Proportion  0.92103 0.93105 0.94032 0.9491 0.95684 0.96361
##                          PC25    PC26    PC27    PC28    PC29    PC30
## Standard deviation     0.4522 0.43655 0.42992 0.39278 0.37190 0.36335
## Proportion of Variance 0.0066 0.00615 0.00596 0.00498 0.00446 0.00426
## Cumulative Proportion  0.9702 0.97636 0.98232 0.98730 0.99176 0.99602
##                           PC31
## Standard deviation     0.35137
## Proportion of Variance 0.00398
## Cumulative Proportion  1.00000
```

```r
# Scree Test: 4 factors
VSS.scree(Data_TEBS, main = "TEBS-Self scree plot")
```

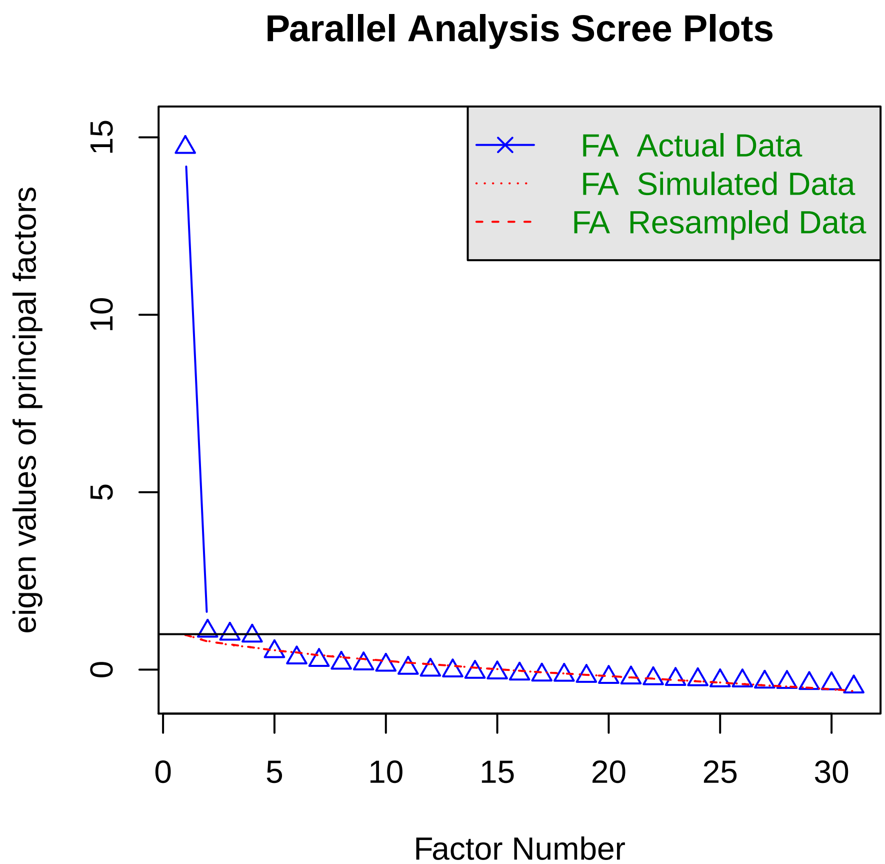

```r
# The Velicer MAP: 4 factors
# BIC: 4 factors
vss(Data_TEBS, fm = "pa")
```




```
##
## Very Simple Structure
## Call: vss(x = Data_TEBS, fm = "pa")
## VSS complexity 1 achieves a maximimum of 0.94  with  1  factors
## VSS complexity 2 achieves a maximimum of 0.95  with  2  factors
##
## The Velicer MAP achieves a minimum of 0.02  with  4  factors
## BIC achieves a minimum of  -1133  with  4  factors
## Sample Size adjusted BIC achieves a minimum of  -150.12  with  8  factors
##
## Statistics by number of factors
##   vss1 vss2   map dof chisq    prob sqresid  fit RMSEA   BIC SABIC complex
## 1 0.94 0.00 0.022 434  1202 2.9e-73    13.7 0.94 0.110 -1014   360     1.0
## 2 0.59 0.95 0.022 404  1027 7.2e-56    11.1 0.95 0.103 -1036   243     1.6
## 3 0.37 0.81 0.020 375   826 5.9e-36     8.9 0.96 0.092 -1089    98     2.1
## 4 0.30 0.66 0.017 347   639 1.5e-19     6.8 0.97 0.078 -1133   -34     2.4
## 5 0.26 0.56 0.018 320   528 2.2e-12     5.8 0.98 0.070 -1106   -93     2.7
## 6 0.27 0.56 0.018 294   440 7.1e-08     5.1 0.98 0.063 -1061  -131     2.9
## 7 0.25 0.52 0.019 269   380 9.6e-06     4.5 0.98 0.059  -994  -142     3.1
## 8 0.20 0.42 0.021 245   325 4.5e-04     4.1 0.98 0.054  -926  -150     3.4
##   eChisq  SRMR eCRMS  eBIC
## 1    829 0.073 0.076 -1387
## 2    613 0.063 0.068 -1449
## 3    419 0.052 0.058 -1496
## 4    240 0.040 0.046 -1532
## 5    169 0.033 0.040 -1465
```

```
## 6     128 0.029 0.036 -1373
## 7     101 0.026 0.034 -1273
## 8      79 0.023 0.031 -1172
```

```r
# Parallel analysis: 4 factors
parallel_TEBS = fa.parallel(Data_TEBS,fm = "pa", fa = "fa")
```




```
## Parallel analysis suggests that the number of factors =  4  and the number of components =  NA
```

```r
## Step 4: TEBS EFA procedure
# fa_TEBS_1 = fa(Data_TEBS, nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_1, digits = 3, cut = 0.45) # 3,4,6,11,12,13,22,23,26,29
#
# fa_TEBS_2 = fa(Data_TEBS[,-c(23)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_2, digits = 3, cut = 0.45) # 3,4,6,11,12,13,22,26,29
#
# fa_TEBS_3 = fa(Data_TEBS[,-c(23,12)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_3, digits = 3, cut = 0.45) # # 3,6,11,13,22,26,29
#
# fa_TEBS_4 = fa(Data_TEBS[,-c(23,12,13)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_4, digits = 3, cut = 0.45) # 3,15,22,29
#
# fa_TEBS_5 = fa(Data_TEBS[,-c(23,12,13,22)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_5, digits = 3, cut = 0.45) # 3,15,29
#
# fa_TEBS_6 = fa(Data_TEBS[,-c(23,12,13,22,29)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_6, digits = 3, cut = 0.45) # 3,11,15
#
```

```r
# fa_TEBS_7 = fa(Data_TEBS[,-c(23,12,13,22,29,3)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_7, digits = 3, cut = 0.45) # 11,15
#
# fa_TEBS_8 = fa(Data_TEBS[,-c(23,12,13,22,29,3,11)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_8, digits = 3, cut = 0.45) # 15
#
# fa_TEBS_9 = fa(Data_TEBS[,-c(23,12,13,22,29,3,11,15)], nfactors = 4, fm = "pa", rotate = "oblimin")
# print(fa_TEBS_9, digits = 3, cut = 0.45)
#
fa_TEBS_FF = fa(Data_TEBS[,-c(23,12,13,22,29,3,11,15)], nfactors = 4, fm = "pa", rotate = "oblimin")
print(fa_TEBS_FF, digits = 3, cut=0.45)
```

```
## Factor Analysis using method =  pa
## Call: fa(r = Data_TEBS[, -c(23, 12, 13, 22, 29, 3, 11, 15)], nfactors = 4,
##     rotate = "oblimin", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##        PA1   PA3   PA4   PA2    h2    u2  com
## x1                         0.497 0.541 0.459 1.79
## x2                         0.488 0.512 0.488 1.53
## x4   0.648                       0.564 0.436 1.35
## x5               0.531           0.564 0.436 1.94
## x6   0.488                       0.565 0.435 1.60
## x7   0.620                       0.576 0.424 1.14
## x8   0.839                       0.657 0.343 1.00
## x9   0.653                       0.611 0.389 1.08
## x10              0.486           0.565 0.435 1.98
## x14        0.544                 0.593 0.407 1.76
## x16              0.493           0.514 0.486 1.51
## x17              0.732           0.701 0.299 1.08
## x18              0.734           0.685 0.315 1.12
## x19        0.551                 0.699 0.301 1.79
## x20        0.701                 0.646 0.354 1.07
## x21        0.831                 0.712 0.288 1.02
## x24  0.753                       0.620 0.380 1.04
## x25        0.814                 0.789 0.211 1.06
## x26  0.477                       0.636 0.364 2.23
## x27                        0.766 0.617 0.383 1.01
## x28                        0.894 0.817 0.183 1.03
## x30  0.618                       0.637 0.363 1.49
## x31  0.708                       0.619 0.381 1.09
##
##                         PA1   PA3   PA4   PA2
## SS loadings           5.143 3.761 2.850 2.686
## Proportion Var        0.224 0.164 0.124 0.117
## Cumulative Var        0.224 0.387 0.511 0.628
## Proportion Explained  0.356 0.260 0.197 0.186
## Cumulative Proportion 0.356 0.617 0.814 1.000
##
##  With factor correlations of
##       PA1   PA3   PA4   PA2
## PA1 1.000 0.611 0.553 0.498
## PA3 0.611 1.000 0.452 0.422
## PA4 0.553 0.452 1.000 0.440
## PA2 0.498 0.422 0.440 1.000
```

```
##
## Mean item complexity =  1.4
## Test of the hypothesis that 4 factors are sufficient.
##
## The degrees of freedom for the null model are  253  and the objective function was  17.583 with Chi S
## The degrees of freedom for the model are 167  and the objective function was  2.183
##
## The root mean square of the residuals (RMSR) is  0.035
## The df corrected root mean square of the residuals is  0.044
##
## The harmonic number of observations is  165 with the empirical chi square  104.366  with prob <  1
## The total number of observations was  165  with Likelihood Chi Square =  333.589  with prob <  3.53e
##
## Tucker Lewis Index of factoring reliability =  0.8963
## RMSEA index =  0.0835  and the 90 % confidence intervals are  0.0658 0.0901
## BIC =  -519.104
## Fit based upon off diagonal values = 0.995
## Measures of factor score adequacy
##                                                     PA1   PA3   PA4   PA2
## Correlation of (regression) scores with factors   0.959 0.955 0.930 0.943
## Multiple R square of scores with factors          0.921 0.911 0.864 0.890
## Minimum correlation of possible factor scores     0.841 0.823 0.729 0.780
```

```r
## Step 5: Summed up items scores
PA1 = c("x4","x6","x7","x8","x9","x24","x26","x30","x31")
PA3 = c("x14","x19","x20","x21","x25")
PA4 = c("x5","x10","x16","x17","x18")
PA2 = c("x1","x2","x27","x28")
# PA1: CM, Management/climate
# PA2: PAID, planning & accommodating for individual differences (Only 14 questioned)
# PA3: HOTS, Higher order thinking skills
# PA4: CC, Communication/Clarification
data$CM <- rowSums(Data_TEBS[PA1])
data$HOTS <- rowSums(Data_TEBS[PA3])
data$CC <- rowSums(Data_TEBS[PA4])
data$PAID <- rowSums(Data_TEBS[PA2])
data$TEBS <- rowSums(Data_TEBS[c(PA1, PA2, PA3, PA4)])
psych::describe(data[c("CM","HOTS","CC","PAID","TEBS")])
```

```
##      vars   n  mean    sd median trimmed   mad min max range  skew
## CM      1 165 26.09  5.86     26   26.35  5.93   9  36    27 -0.34
## HOTS    2 165 13.90  3.52     14   13.89  4.45   5  20    15 -0.05
## CC      3 165 14.81  3.16     15   14.86  2.97   5  20    15 -0.16
## PAID    4 165 10.38  2.59     10   10.26  2.97   4  16    12  0.32
## TEBS    5 165 65.18 13.16     65   65.36 13.34  23  92    69 -0.19
##      kurtosis   se
## CM      -0.42 0.46
## HOTS    -0.62 0.27
## CC      -0.47 0.25
## PAID    -0.15 0.20
## TEBS    -0.17 1.02
```

```r
# mvn(data[c("CM","HOTS","CC","PAID","TEBS")], multivariatePlot = "qq")

## Step 6: Reliability Test
```

```r
alpha.pa1 = psych::alpha(Data_TEBS[PA1]) # CM
print(alpha.pa1$total, digits = 3) # 0.926
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##      0.926     0.927    0.93     0.585 12.7 0.0086  2.9 0.651    0.582
```

```r
alpha.pa2 = psych::alpha(Data_TEBS[PA2]) # PAID
print(alpha.pa2$total, digits = 3) # 0.846
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##      0.846     0.846   0.835     0.578 5.49 0.0201  2.6 0.647     0.55
```

```r
alpha.pa3 = psych::alpha(Data_TEBS[PA3]) # HOTS
print(alpha.pa3$total, digits = 3) # 0.901
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##      0.901     0.901   0.885     0.647 9.15 0.0123 2.78 0.705    0.633
```

```r
alpha.pa4 = psych::alpha(Data_TEBS[PA4]) # CC
print(alpha.pa4$total, digits = 3) # 0.869
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N    ase mean    sd median_r
##      0.869     0.869   0.859     0.571 6.66 0.0161 2.96 0.632    0.559
```

```r
alpha.patial_all = psych::alpha(Data_TEBS[c(PA1,PA2,PA3,PA4)]) # all 23 items
print(alpha.patial_all$total, digits = 3) # 0.955
```

```
##  raw_alpha std.alpha G6(smc) average_r  S/N     ase mean    sd median_r
##      0.955     0.955    0.97     0.479 21.1 0.00512 2.83 0.572    0.469
```

3. Aggregate Analysis of Teacher CoP

3.1. Network Description

As mentioned in the article, we need to calculate, on the average, overall frequencies of interaction and levels of energizing contacts of each teacher.

```r
network_attr <- network[c("networksize_overall","density_overall","bridge_overall",
                          "reach_overall","totalfreq_overall","totalenergize_overall")]
network_attr$avgfreq_overall = network_attr$totalfreq_overall/network_attr$networksize_overall
network_attr$avgenergize_overall = network_attr$totalenergize_overall/network_attr$networksize_overall

psych::describe(network_attr)
```

```
##                         vars   n  mean    sd median trimmed   mad  min max
## networksize_overall        1 165  9.84  6.42   9.00    9.05  4.45 0.00  38
## density_overall            2 165  3.04  0.76   3.00    3.06  1.48 1.00   4
## bridge_overall             3 165  2.68  1.00   3.00    2.65  1.48 1.00   5
## reach_overall              4 165  2.80  0.85   3.00    2.83  1.48 1.00   4
## totalfreq_overall          5 165 33.67 19.98  31.00   32.01 17.79 0.00 124
## totalenergize_overall      6 165 39.88 26.76  35.00   36.77 22.24 0.00 155
## avgfreq_overall            7 159  3.56  0.71   3.56    3.56  0.65 1.00   5
## avgenergize_overall        8 159  4.01  0.62   4.07    4.04  0.53 1.75   5
##                          range  skew kurtosis   se
## networksize_overall      38.00  1.59     3.89 0.50
## density_overall           3.00 -0.22    -0.84 0.06
```

```
## bridge_overall          4.00  0.42    -0.32 0.08
## reach_overall           3.00 -0.26    -0.59 0.07
## totalfreq_overall     124.00  1.30     3.17 1.56
## totalenergize_overall 155.00  1.35     2.57 2.08
## avgfreq_overall         4.00 -0.25     0.39 0.06
## avgenergize_overall     3.25 -0.79     1.17 0.05
```

3.2 Network Plots: Figure 2

Fig2.A Geography

```r
sum(network_plot$networksize_school, na.rm = T) # 855
```

```
## [1] 855
```

```r
sum(network_plot$networksize_district, na.rm = T) # 350
```

```
## [1] 350
```

```r
sum(network_plot$networksize_state, na.rm = T) # 278
```

```
## [1] 278
```

```r
sum(network_plot$networksize_nation, na.rm = T) # 140
```

```
## [1] 140
```

```r
counts_1 <- c(855,350,278,140)
sum <- sum(counts_1)
counts_1 <- round(counts_1*100/sum, 2)
xx <- barplot(counts_1, main="(A) Geograpy", names.arg = c("School","District","State","Nation"),
              xlab="Geographic Proximity to Teacher", ylab = "All reported contacts (%)",
              ylim = c(0, 1.3*max(counts_1)), col = gray.colors(5))
text(x = xx, y = counts_1, label = counts_1, pos = 3, cex = 1.2, col = "black")
```

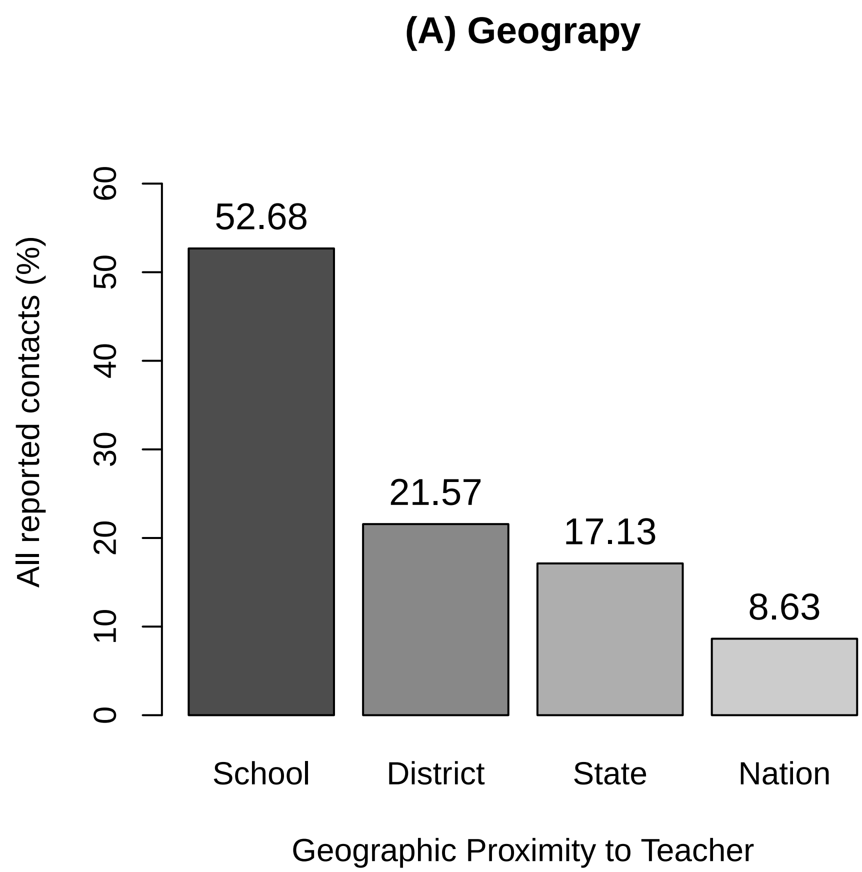


Fig2.B Reach

```
counts_4 <- table(network_plot$reach_overall)
xx <- barplot(counts_4, main="(B) Overall Reach",
              xlab="Respond", ylab = "Number of Participants",
              ylim = c(0, 1.3*max(counts_4)), col = gray.colors(5))
text(x = xx, y = counts_4, label = counts_4, pos = 3, cex = 1.2, col = "black")
```

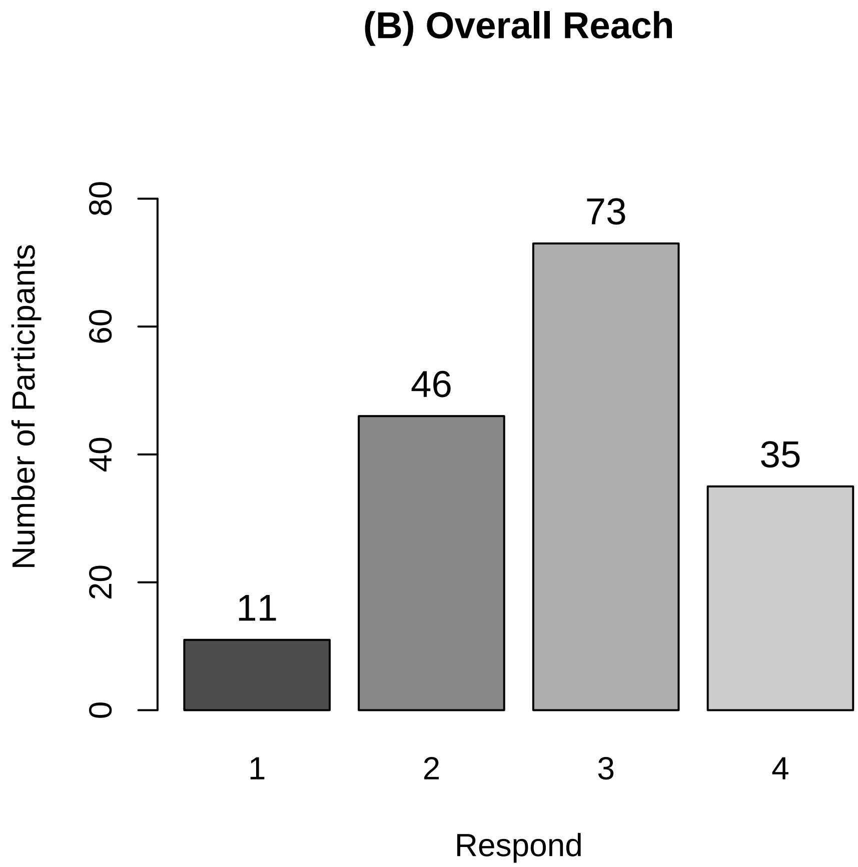


Fig2.C Density

```
(counts_2 <- table(network_plot$density_overall))
```

```
##
##  1  2  3  4
##  2 39 75 49
```

```
counts_2 <- c(2,39,75,49,0)
counts_2 <- as.table(counts_2)
names(counts_2) <- c("1","2","3","4","5")
xx <- barplot(counts_2, main="(C) Overall Density",
              xlab="Respond", ylab = "Number of Participants",
              ylim = c(0, 1.3*max(counts_2)), col = gray.colors(5))
text(x = xx, y = counts_2, label = counts_2, pos = 3, cex = 1.2, col = "black")
```

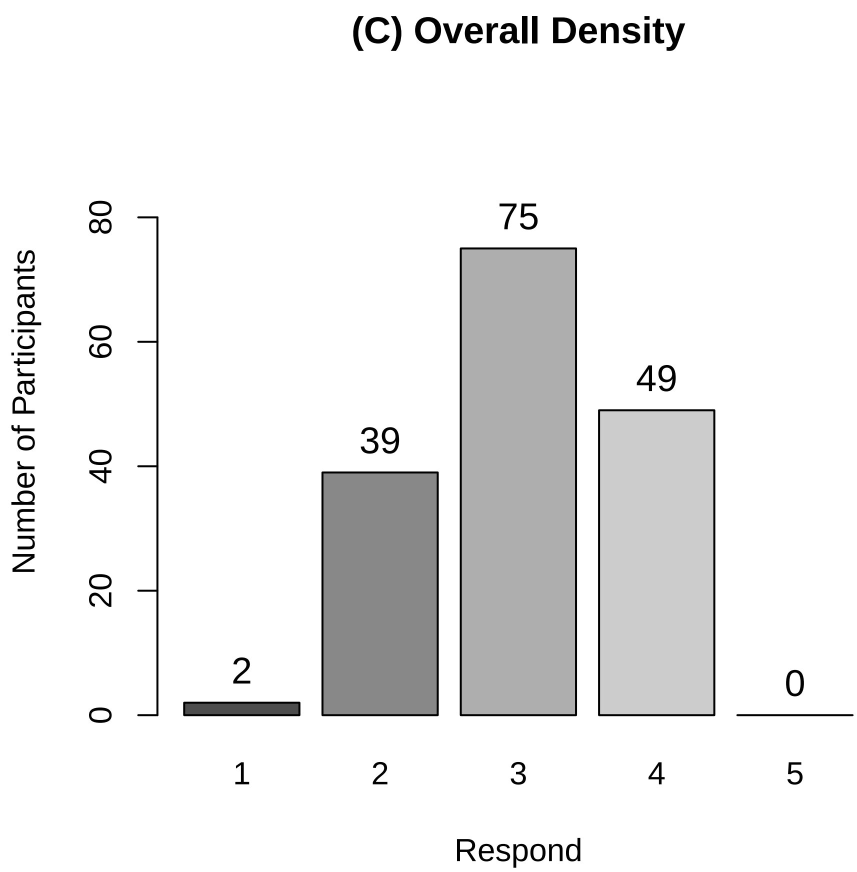
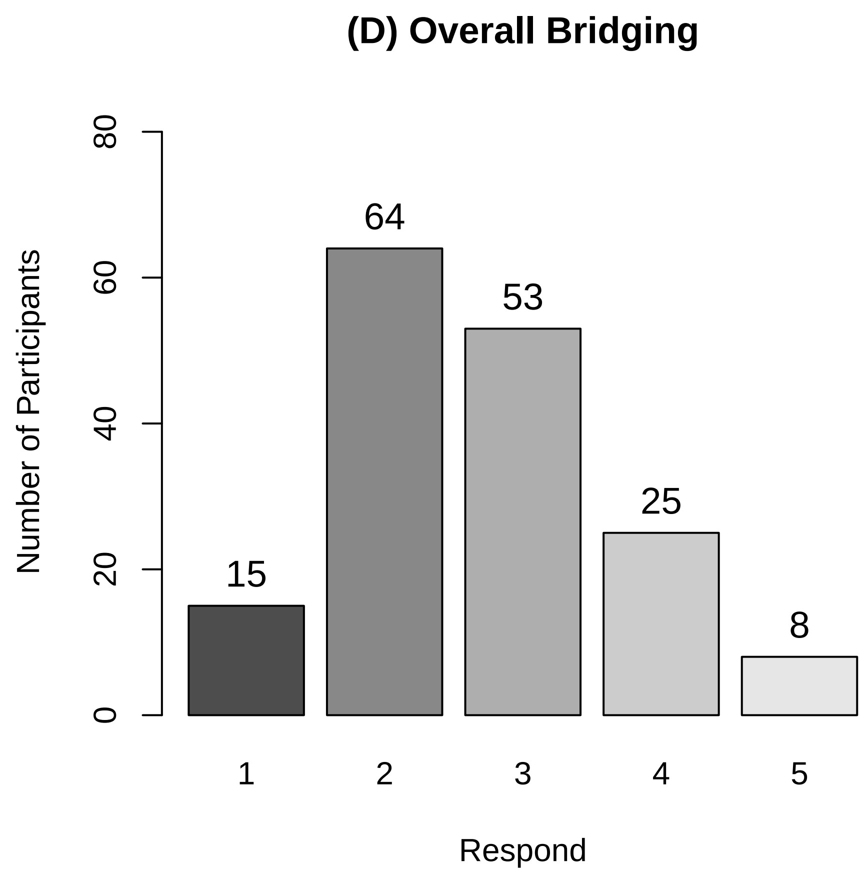


Fig2.D Bridging

```
counts_3 <- table(network_plot$bridge_overall)
xx <- barplot(counts_3, main="(D) Overall Bridging",
              xlab="Respond", ylab = "Number of Participants",
              ylim = c(0, 1.3*max(counts_3)), col = gray.colors(5))
text(x = xx, y = counts_3, label = counts_3, pos = 3, cex = 1.2, col = "black")
```



4. Correlation Analysis

Use Listwise_deletion, and calculate the spearman correlation.

```
corrdata_final <- na.omit(corrdata)
rs_s_final = rcorr(as.matrix(corrdata_final[,-1], type="spearman"))
print(rs_s_final,digits=2)
```

```
##                      SMTI    SI    CA    SR   CV  TEBS    CM HOTS PAID
## SMTI                 1.00  0.87  0.79  0.51 0.62  0.55  0.51 0.44 0.43
## SI                   0.87  1.00  0.49  0.23 0.40  0.53  0.54 0.45 0.41
## CA                   0.79  0.49  1.00  0.31 0.47  0.42  0.36 0.34 0.30
## SR                   0.51  0.23  0.31  1.00 0.27  0.17  0.10 0.12 0.21
## CV                   0.62  0.40  0.47  0.27 1.00  0.32  0.29 0.22 0.24
## TEBS                 0.55  0.53  0.42  0.17 0.32  1.00  0.93 0.87 0.77
## CM                   0.51  0.54  0.36  0.10 0.29  0.93  1.00 0.72 0.62
## HOTS                 0.44  0.45  0.34  0.12 0.22  0.87  0.72 1.00 0.60
## PAID                 0.43  0.41  0.30  0.21 0.24  0.77  0.62 0.60 1.00
## CC                   0.48  0.38  0.44  0.20 0.35  0.84  0.70 0.68 0.56
## networksize_overall  0.23  0.15  0.33  0.03 0.14  0.12  0.11 0.06 0.10
## density_overall      0.08  0.07  0.11 -0.06 0.05  0.08  0.08 0.07 0.06
## bridge_overall       0.31  0.23  0.33  0.14 0.18  0.21  0.19 0.20 0.25
## reach_overall        0.24  0.24  0.22 -0.03 0.18  0.11  0.08 0.12 0.11
## avgfreq_overall     -0.13 -0.10 -0.16 -0.09 0.01 -0.04 -0.03 0.01 0.00
## avgenergize_overall  0.33  0.28  0.32  0.06 0.26  0.19  0.19 0.13 0.14
```

```
##                         CC networksize_overall density_overall
## SMTI                  0.48                0.23            0.08
## SI                    0.38                0.15            0.07
## CA                    0.44                0.33            0.11
## SR                    0.20                0.03           -0.06
## CV                    0.35                0.14            0.05
## TEBS                  0.84                0.12            0.08
## CM                    0.70                0.11            0.08
## HOTS                  0.68                0.06            0.07
## PAID                  0.56                0.10            0.06
## CC                    1.00                0.13            0.04
## networksize_overall   0.13                1.00            0.07
## density_overall       0.04                0.07            1.00
## bridge_overall        0.11                0.29            0.05
## reach_overall         0.09                0.21            0.29
## avgfreq_overall      -0.14               -0.31            0.03
## avgenergize_overall   0.18                0.13            0.16
##                     bridge_overall reach_overall avgfreq_overall
## SMTI                          0.31          0.24           -0.13
## SI                            0.23          0.24           -0.10
## CA                            0.33          0.22           -0.16
## SR                            0.14         -0.03           -0.09
## CV                            0.18          0.18            0.01
## TEBS                          0.21          0.11           -0.04
## CM                            0.19          0.08           -0.03
## HOTS                          0.20          0.12            0.01
## PAID                          0.25          0.11            0.00
## CC                            0.11          0.09           -0.14
## networksize_overall           0.29          0.21           -0.31
## density_overall               0.05          0.29            0.03
## bridge_overall                1.00          0.11           -0.11
## reach_overall                 0.11          1.00           -0.20
## avgfreq_overall              -0.11         -0.20            1.00
## avgenergize_overall           0.10          0.18            0.14
##                     avgenergize_overall
## SMTI                               0.33
## SI                                 0.28
## CA                                 0.32
## SR                                 0.06
## CV                                 0.26
## TEBS                               0.19
## CM                                 0.19
## HOTS                               0.13
## PAID                               0.14
## CC                                 0.18
## networksize_overall                0.13
## density_overall                    0.16
## bridge_overall                     0.10
## reach_overall                      0.18
## avgfreq_overall                    0.14
## avgenergize_overall                1.00
##
## n= 159
##
```

```
##
## P
##                     SMTI   SI     CA     SR     CV     TEBS   CM
## SMTI                       0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
## SI                  0.0000        0.0000 0.0029 0.0000 0.0000 0.0000
## CA                  0.0000 0.0000        0.0000 0.0000 0.0000 0.0000
## SR                  0.0000 0.0029 0.0000        0.0005 0.0344 0.1912
## CV                  0.0000 0.0000 0.0000 0.0005        0.0000 0.0002
## TEBS                0.0000 0.0000 0.0000 0.0344 0.0000        0.0000
## CM                  0.0000 0.0000 0.0000 0.1912 0.0002 0.0000
## HOTS                0.0000 0.0000 0.0000 0.1290 0.0062 0.0000 0.0000
## PAID                0.0000 0.0000 0.0001 0.0089 0.0022 0.0000 0.0000
## CC                  0.0000 0.0000 0.0000 0.0109 0.0000 0.0000 0.0000
## networksize_overall 0.0031 0.0674 0.0000 0.6943 0.0714 0.1448 0.1652
## density_overall     0.3460 0.3692 0.1803 0.4599 0.5610 0.3376 0.2978
## bridge_overall      0.0000 0.0036 0.0000 0.0771 0.0196 0.0065 0.0160
## reach_overall       0.0020 0.0023 0.0053 0.7384 0.0201 0.1627 0.3168
## avgfreq_overall     0.0932 0.2132 0.0476 0.2381 0.8839 0.5920 0.7105
## avgenergize_overall 0.0000 0.0003 0.0000 0.4623 0.0010 0.0161 0.0185
##                     HOTS   PAID   CC     networksize_overall
## SMTI                0.0000 0.0000 0.0000 0.0031
## SI                  0.0000 0.0000 0.0000 0.0674
## CA                  0.0000 0.0001 0.0000 0.0000
## SR                  0.1290 0.0089 0.0109 0.6943
## CV                  0.0062 0.0022 0.0000 0.0714
## TEBS                0.0000 0.0000 0.0000 0.1448
## CM                  0.0000 0.0000 0.0000 0.1652
## HOTS                       0.0000 0.0000 0.4463
## PAID                0.0000        0.0000 0.2299
## CC                  0.0000 0.0000        0.0946
## networksize_overall 0.4463 0.2299 0.0946
## density_overall     0.4147 0.4707 0.5754 0.3851
## bridge_overall      0.0123 0.0012 0.1673 0.0002
## reach_overall       0.1464 0.1517 0.2487 0.0069
## avgfreq_overall     0.9134 0.9653 0.0843 0.0000
## avgenergize_overall 0.0976 0.0691 0.0210 0.1103
##                     density_overall bridge_overall reach_overall
## SMTI                0.3460          0.0000         0.0020
## SI                  0.3692          0.0036         0.0023
## CA                  0.1803          0.0000         0.0053
## SR                  0.4599          0.0771         0.7384
## CV                  0.5610          0.0196         0.0201
## TEBS                0.3376          0.0065         0.1627
## CM                  0.2978          0.0160         0.3168
## HOTS                0.4147          0.0123         0.1464
## PAID                0.4707          0.0012         0.1517
## CC                  0.5754          0.1673         0.2487
## networksize_overall 0.3851          0.0002         0.0069
## density_overall                     0.5180         0.0002
## bridge_overall      0.5180                         0.1629
## reach_overall       0.0002          0.1629
## avgfreq_overall     0.7241          0.1820         0.0125
## avgenergize_overall 0.0398          0.2029         0.0265
##                     avgfreq_overall avgenergize_overall
```

```
## SMTI                 0.0932           0.0000
## SI                   0.2132           0.0003
## CA                   0.0476           0.0000
## SR                   0.2381           0.4623
## CV                   0.8839           0.0010
## TEBS                 0.5920           0.0161
## CM                   0.7105           0.0185
## HOTS                 0.9134           0.0976
## PAID                 0.9653           0.0691
## CC                   0.0843           0.0210
## networksize_overall 0.0000           0.1103
## density_overall      0.7241           0.0398
## bridge_overall       0.1820           0.2029
## reach_overall        0.0125           0.0265
## avgfreq_overall                       0.0819
## avgenergize_overall 0.0819
```

```
# rs_s_final$P
```

5. Comparison of Science and Math Teachers

```
psych::describe(data_math)
```

```
##                     vars  n  mean    sd median trimmed   mad  min max
## SMTI                   1 36 42.89  7.50  44.00   43.73  5.93 13.0  52
## SI                     2 36 19.56  4.67  21.00   20.23  2.97  6.0  24
## CA                     3 36  9.25  2.25   9.00    9.50  2.97  3.0  12
## SR                     4 36  6.67  1.43   7.00    6.87  1.48  2.0   8
## CV                     5 36  7.42  1.16   8.00    7.63  0.00  2.0   8
## TEBS                   6 36 62.67 12.36  61.50   62.87 11.86 23.0  87
## CM                     7 36 24.61  5.94  25.00   24.90  5.93  9.0  34
## HOTS                   8 36 12.86  3.45  13.00   12.73  2.97  5.0  20
## PAID                   9 36 10.56  2.50  10.50   10.53  2.22  4.0  16
## CC                    10 36 14.64  3.03  14.00   14.70  1.48  5.0  20
## networksize_overall   11 36  8.00  4.74   7.00    7.57  2.97  0.0  25
## density_overall       12 36  3.03  0.74   3.00    3.07  0.00  1.0   4
## bridge_overall        13 36  2.31  0.89   2.00    2.23  0.00  1.0   5
## reach_overall         14 36  2.69  0.79   3.00    2.70  1.48  1.0   4
## avgfreq_overall       15 34  3.71  0.73   3.70    3.75  0.61  1.6   5
## avgenergize_overall   16 34  4.07  0.52   4.15    4.10  0.43  2.0   5
##                     range  skew kurtosis   se
## SMTI                 39.0 -1.77     4.68 1.25
## SI                   18.0 -1.36     1.17 0.78
## CA                    9.0 -0.88     0.40 0.37
## SR                    6.0 -1.29     1.73 0.24
## CV                    6.0 -2.99    10.70 0.19
## TEBS                 64.0 -0.53     1.17 2.06
## CM                   25.0 -0.40    -0.24 0.99
## HOTS                 15.0  0.19    -0.34 0.57
## PAID                 12.0 -0.02    -0.10 0.42
## CC                   15.0 -0.35     1.39 0.50
## networksize_overall  25.0  1.32     2.74 0.79
```

```
## density_overall       3.0 -0.46    -0.01 0.12
## bridge_overall        4.0  0.82     0.83 0.15
## reach_overall         3.0 -0.12    -0.53 0.13
## avgfreq_overall       3.4 -0.68     0.71 0.12
## avgenergize_overall   3.0 -1.59     5.26 0.09
```

```r
psych::describe(data_science)
```

```
##                     vars   n  mean    sd median trimmed   mad   min max
## SMTI                   1 129 45.14  5.40   47.0   45.71  4.45 31.00  52
## SI                     2 129 21.11  3.11   22.0   21.57  2.97 10.00  24
## CA                     3 129  9.75  2.17   10.0   10.03  2.97  3.00  12
## SR                     4 129  6.83  1.17    7.0    6.97  1.48  2.00   8
## CV                     5 129  7.45  0.85    8.0    7.59  0.00  5.00   8
## TEBS                   6 129 65.88 13.33   65.0   66.08 14.83 32.00  92
## CM                     7 129 26.50  5.80   26.0   26.76  5.93 12.00  36
## HOTS                   8 129 14.19  3.50   14.0   14.23  4.45  5.00  20
## PAID                   9 129 10.33  2.62   10.0   10.20  2.97  4.00  16
## CC                    10 129 14.86  3.20   15.0   14.90  2.97  8.00  20
## networksize_overall   11 129 10.35  6.74    9.0    9.55  4.45  0.00  38
## density_overall       12 129  3.04  0.77    3.0    3.06  1.48  1.00   4
## bridge_overall        13 129  2.78  1.01    3.0    2.76  1.48  1.00   5
## reach_overall         14 129  2.83  0.87    3.0    2.88  1.48  1.00   4
## avgfreq_overall       15 125  3.51  0.71    3.5    3.51  0.62  1.00   5
## avgenergize_overall   16 125  3.99  0.64    4.0    4.02  0.59  1.75   5
##                     range  skew kurtosis   se
## SMTI                21.00 -0.88    -0.10 0.48
## SI                  14.00 -1.25     1.08 0.27
## CA                   9.00 -0.94     0.34 0.19
## SR                   6.00 -0.97     1.12 0.10
## CV                   3.00 -1.30     0.49 0.07
## TEBS                60.00 -0.14    -0.55 1.17
## CM                  24.00 -0.32    -0.57 0.51
## HOTS                15.00 -0.13    -0.67 0.31
## PAID                12.00  0.41    -0.17 0.23
## CC                  12.00 -0.11    -0.93 0.28
## networksize_overall 38.00  1.52     3.44 0.59
## density_overall      3.00 -0.17    -1.07 0.07
## bridge_overall       4.00  0.30    -0.46 0.09
## reach_overall        3.00 -0.31    -0.62 0.08
## avgfreq_overall      4.00 -0.13     0.39 0.06
## avgenergize_overall  3.25 -0.63     0.58 0.06
```

5.1 TEBS

```r
wilcox.test(data_math$CM,data_science$CM) # 0.1099
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$CM and data_science$CM
## W = 1917, p-value = 0.1099
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$HOTS,data_science$HOTS) # 0.04437
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$HOTS and data_science$HOTS
## W = 1814, p-value = 0.04437
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$PAID,data_science$PAID) # 0.485
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$PAID and data_science$PAID
## W = 2498, p-value = 0.485
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$CC,data_science$CC) # 0.6655
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$CC and data_science$CC
## W = 2212.5, p-value = 0.6655
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$TEBS,data_science$TEBS) # 0.2116
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$TEBS and data_science$TEBS
## W = 2005, p-value = 0.2116
## alternative hypothesis: true location shift is not equal to 0
```

5.2 SMTI

```r
wilcox.test(data_math$SI,data_science$SI) # 0.08565
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$SI and data_science$SI
## W = 1891.5, p-value = 0.08565
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$CA,data_science$CA) # 0.1794
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$CA and data_science$CA
## W = 1986.5, p-value = 0.1794
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$SR,data_science$SR) # 0.7058
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$SR and data_science$SR
## W = 2230, p-value = 0.7058
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$CV,data_science$CV) # 0.8078
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$CV and data_science$CV
## W = 2374.5, p-value = 0.8078
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$SMTI,data_science$SMTI) # 0.08148
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$SMTI and data_science$SMTI
## W = 1881, p-value = 0.08148
## alternative hypothesis: true location shift is not equal to 0
```

5.3. Network

```r
wilcox.test(data_math$networksize_overall,data_science$networksize_overall) # 0.0335
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$networksize_overall and data_science$networksize_overall
## W = 1784, p-value = 0.0335
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$avgfreq_overall,data_science$avgfreq_overall) # 0.09407
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$avgfreq_overall and data_science$avgfreq_overall
## W = 2524, p-value = 0.09407
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$avgenergize_overall,data_science$avgenergize_overall) # 0.552
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$avgenergize_overall and data_science$avgenergize_overall
## W = 2267, p-value = 0.552
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$density_overall,data_science$density_overall) # 0.9831
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$density_overall and data_science$density_overall
## W = 2316.5, p-value = 0.9831
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$bridge_overall,data_science$bridge_overall) # 0.007969
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$bridge_overall and data_science$bridge_overall
## W = 1682, p-value = 0.007969
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(data_math$reach_overall,data_science$reach_overall) # 0.3477
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_math$reach_overall and data_science$reach_overall
## W = 2098, p-value = 0.3477
## alternative hypothesis: true location shift is not equal to 0
```

5.4. SMTI_details

```r
data_SMTI <- merge(STI, science, by = "id")
data_SMTI_math <- data_SMTI[data_SMTI$Math_or_Science==1,]
data_SMTI_science <- data_SMTI[data_SMTI$Math_or_Science==2,]
psych::describe(data_SMTI_math)
```

```
##                   vars  n mean   sd median trimmed  mad min  max range  skew
## id*                  1 36  NaN   NA     NA     NaN   NA Inf -Inf  -Inf    NA
## x1                   2 36 3.53 0.70    4.0    3.63 0.00   1    4     3 -1.58
## x2                   3 36 3.50 0.70    4.0    3.60 0.00   1    4     3 -1.48
## x3                   4 36 3.22 0.87    3.0    3.33 1.48   1    4     3 -0.94
## x4                   5 36 2.53 1.16    2.5    2.53 2.22   1    4     3 -0.01
## x5                   6 36 3.50 0.77    4.0    3.63 0.00   1    4     3 -1.79
## x6                   7 36 3.69 0.62    4.0    3.80 0.00   1    4     3 -2.45
## x7                   8 36 3.72 0.61    4.0    3.83 0.00   1    4     3 -2.66
## x8                   9 36 2.92 0.81    3.0    2.97 0.00   1    4     3 -0.49
## x9                  10 36 3.36 0.72    3.0    3.47 1.48   1    4     3 -1.07
## x10                 11 36 3.42 0.84    4.0    3.57 0.00   1    4     3 -1.43
## x11                 12 36 3.25 0.77    3.0    3.33 1.48   1    4     3 -0.80
## x12                 13 36 3.56 0.69    4.0    3.67 0.00   1    4     3 -1.68
## x13                 14 36 3.03 0.84    3.0    3.10 1.48   1    4     3 -0.60
## x14                 15 36 3.31 0.86    3.5    3.43 0.74   1    4     3 -1.13
## x15                 16 36 3.28 0.97    4.0    3.43 0.00   1    4     3 -1.10
## x16                 17 36 3.33 0.93    4.0    3.47 0.00   1    4     3 -1.10
## x17                 18 36 3.19 0.95    3.0    3.33 1.48   1    4     3 -0.96
## x18                 19 36 3.42 0.84    4.0    3.57 0.00   1    4     3 -1.43
```

```
## Math_or_Science   20 36 1.00 0.00    1.0    1.00 0.00   1    1     0   NaN
##                 kurtosis   se
## id*                   NA   NA
## x1                  2.71 0.12
## x2                  2.45 0.12
## x3                  0.15 0.14
## x4                 -1.50 0.19
## x5                  3.19 0.13
## x6                  7.03 0.10
## x7                  8.09 0.10
## x8                 -0.18 0.13
## x9                  1.17 0.12
## x10                 1.39 0.14
## x11                 0.18 0.13
## x12                 3.02 0.12
## x13                -0.25 0.14
## x14                 0.60 0.14
## x15                -0.01 0.16
## x16                 0.00 0.15
## x17                -0.11 0.16
## x18                 1.39 0.14
## Math_or_Science      NaN 0.00
```

```
psych::describe(data_SMTI_science)
```

```
##                 vars   n mean   sd median trimmed  mad min  max range
## id*                1 129  NaN   NA     NA     NaN   NA Inf -Inf  -Inf
## x1                 2 129 3.63 0.50      4    3.67 0.00   2    4     2
## x2                 3 129 3.43 0.66      4    3.52 0.00   1    4     3
## x3                 4 129 3.24 0.84      3    3.33 1.48   1    4     3
## x4                 5 129 3.01 1.00      3    3.12 1.48   1    4     3
## x5                 6 129 3.50 0.73      4    3.64 0.00   1    4     3
## x6                 7 129 3.71 0.52      4    3.80 0.00   2    4     2
## x7                 8 129 3.74 0.46      4    3.80 0.00   2    4     2
## x8                 9 129 3.08 0.77      3    3.14 0.00   1    4     3
## x9                10 129 3.50 0.59      4    3.56 0.00   2    4     2
## x10               11 129 3.57 0.61      4    3.65 0.00   1    4     3
## x11               12 129 3.26 0.69      3    3.33 1.48   1    4     3
## x12               13 129 3.44 0.66      4    3.54 0.00   2    4     2
## x13               14 129 3.12 0.84      3    3.22 1.48   1    4     3
## x14               15 129 3.49 0.64      4    3.58 0.00   2    4     2
## x15               16 129 3.66 0.62      4    3.78 0.00   1    4     3
## x16               17 129 3.68 0.62      4    3.82 0.00   1    4     3
## x17               18 129 3.54 0.68      4    3.68 0.00   1    4     3
## x18               19 129 3.61 0.58      4    3.70 0.00   2    4     2
## Math_or_Science   20 129 2.00 0.00      2    2.00 0.00   2    2     0
##                  skew kurtosis   se
## id*                NA       NA   NA
## x1              -0.71    -1.06 0.04
## x2              -0.89     0.35 0.06
## x3              -0.79    -0.30 0.07
## x4              -0.66    -0.71 0.09
## x5              -1.33     1.01 0.06
## x6              -1.58     1.58 0.05
## x7              -1.30     0.29 0.04
```

```
## x8              -0.65     0.25 0.07
## x9              -0.70    -0.51 0.05
## x10             -1.27     1.50 0.05
## x11             -0.53    -0.28 0.06
## x12             -0.76    -0.54 0.06
## x13             -0.79     0.09 0.07
## x14             -0.85    -0.36 0.06
## x15             -1.79     2.79 0.05
## x16             -1.95     3.18 0.06
## x17             -1.31     0.90 0.06
## x18             -1.17     0.34 0.05
## Math_or_Science   NaN      NaN 0.00
```

```r
wilcox.test(data_SMTI_math$x13,data_SMTI_science$x13, "less") # 0.248
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x13 and data_SMTI_science$x13
## W = 2161.5, p-value = 0.248
## alternative hypothesis: true location shift is less than 0
```

```r
wilcox.test(data_SMTI_math$x14,data_SMTI_science$x14, "less") # 0.1722
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x14 and data_SMTI_science$x14
## W = 2109, p-value = 0.1722
## alternative hypothesis: true location shift is less than 0
```

```r
wilcox.test(data_SMTI_math$x15,data_SMTI_science$x15, "less") # 0.01142
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x15 and data_SMTI_science$x15
## W = 1853, p-value = 0.01142
## alternative hypothesis: true location shift is less than 0
```

```r
wilcox.test(data_SMTI_math$x16,data_SMTI_science$x16, "less") # 0.01115
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x16 and data_SMTI_science$x16
## W = 1866, p-value = 0.01115
## alternative hypothesis: true location shift is less than 0
```

```r
wilcox.test(data_SMTI_math$x17,data_SMTI_science$x17, "less") # 0.01868
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x17 and data_SMTI_science$x17
## W = 1863, p-value = 0.01868
## alternative hypothesis: true location shift is less than 0
```

```r
wilcox.test(data_SMTI_math$x18,data_SMTI_science$x18, "less") # 0.1469
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data_SMTI_math$x18 and data_SMTI_science$x18
## W = 2097.5, p-value = 0.1469
## alternative hypothesis: true location shift is less than 0
```

6. Appendix Material

6.1 Higher Order Factor Model

TEBS

```r
TEBS_model_fi <- '
CM =~ x4 + x6 + x7 + x8 + x9 + x24 + x26 + x30 + x31
HOTS =~ x14 + x19 + x20 + x21 + x25
PAID =~ x1 + x2 + x27 + x28
CC =~ x5 + x10 + x16 + x17 + x18
'
TEBS_model_se <- '
CM =~ x4 + x6 + x7 + x8 + x9 + x24 + x26 + x30 + x31
HOTS =~ x14 + x19 + x20 + x21 + x25
PAID =~ x1 + x2 + x27 + x28
CC =~ x5 + x10 + x16 + x17 + x18
TEBS =~ CM + HOTS + PAID + CC
'
```

```r
fit_TEBS<- cfa(TEBS_model_se,
               data=Data_TEBS,estimator = "MLR", std.lv = TRUE)
summary(fit_TEBS, standardized = TRUE,
        fit.measures = TRUE)
```

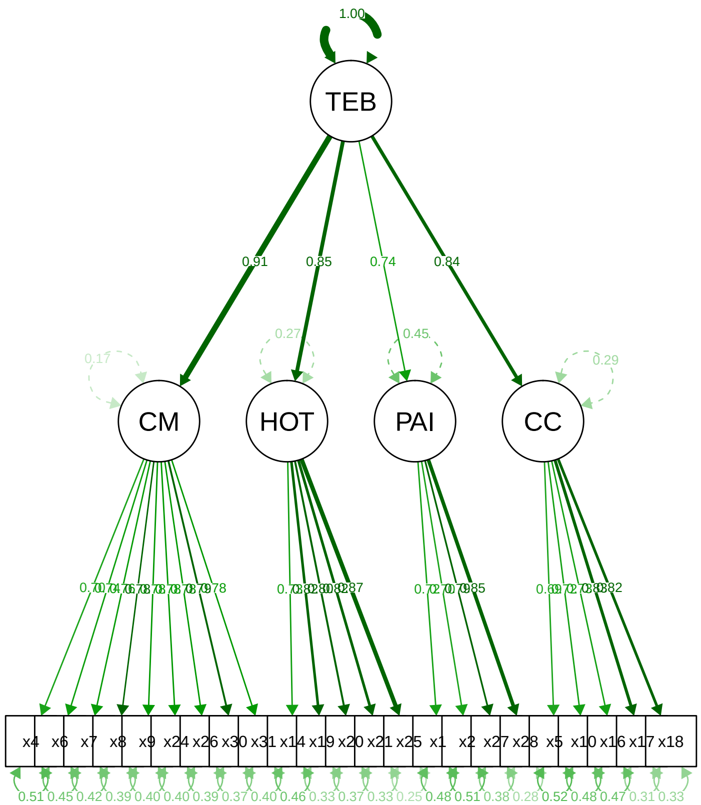
```
## lavaan 0.6-5 ended normally after 72 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         50
##
##   Number of observations                           165
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               546.841     507.391
##   Degrees of freedom                               226         226
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.078
##     for the Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
```

```
##   Test statistic                              2901.125    2642.448
##   Degrees of freedom                               253         253
##   P-value                                        0.000       0.000
##   Scaling correction factor                                  1.098
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.879       0.882
##   Tucker-Lewis Index (TLI)                       0.864       0.868
##
##   Robust Comparative Fit Index (CFI)                         0.884
##   Robust Tucker-Lewis Index (TLI)                            0.871
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -3375.639   -3375.639
##   Scaling correction factor                                  1.020
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)      -3102.219   -3102.219
##   Scaling correction factor                                  1.067
##       for the MLR correction
##
##   Akaike (AIC)                                6851.279    6851.279
##   Bayesian (BIC)                              7006.576    7006.576
##   Sample-size adjusted Bayesian (BIC)         6848.276    6848.276
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.093       0.087
##   90 Percent confidence interval - lower         0.083       0.077
##   90 Percent confidence interval - upper         0.103       0.097
##   P-value RMSEA <= 0.05                          0.000       0.000
##
##   Robust RMSEA                                               0.090
##   90 Percent confidence interval - lower                     0.080
##   90 Percent confidence interval - upper                     0.101
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.069       0.069
##
## Parameter Estimates:
##
##   Information                                 Observed
##   Observed information based on                Hessian
##   Standard errors                    Robust.huber.white
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   CM =~
##     x4                0.227    0.036    6.393    0.000    0.550    0.699
##     x6                0.254    0.040    6.437    0.000    0.615    0.743
##     x7                0.267    0.042    6.325    0.000    0.646    0.758
##     x8                0.290    0.051    5.673    0.000    0.701    0.780
```

```
##     x9                0.233    0.040    5.867    0.000    0.563    0.778
##     x24               0.285    0.047    6.086    0.000    0.690    0.777
##     x26               0.256    0.041    6.218    0.000    0.620    0.780
##     x30               0.257    0.042    6.175    0.000    0.622    0.794
##     x31               0.255    0.044    5.825    0.000    0.618    0.777
##   HOTS =~
##     x14               0.318    0.037    8.524    0.000    0.611    0.734
##     x19               0.366    0.043    8.475    0.000    0.703    0.816
##     x20               0.346    0.047    7.319    0.000    0.665    0.796
##     x21               0.354    0.048    7.437    0.000    0.680    0.818
##     x25               0.356    0.047    7.654    0.000    0.685    0.866
##   PAID =~
##     x1                0.371    0.042    8.882    0.000    0.553    0.718
##     x2                0.372    0.040    9.208    0.000    0.555    0.703
##     x27               0.419    0.097    4.323    0.000    0.625    0.788
##     x28               0.437    0.093    4.676    0.000    0.652    0.848
##   CC =~
##     x5                0.270    0.035    7.696    0.000    0.500    0.691
##     x10               0.315    0.040    7.899    0.000    0.582    0.723
##     x16               0.282    0.035    8.104    0.000    0.522    0.726
##     x17               0.370    0.058    6.384    0.000    0.685    0.829
##     x18               0.356    0.057    6.263    0.000    0.658    0.820
##   TEBS =~
##     CM                2.204    0.416    5.300    0.000    0.911    0.911
##     HOTS              1.641    0.286    5.743    0.000    0.854    0.854
##     PAID              1.107    0.294    3.764    0.000    0.742    0.742
##     CC                1.556    0.281    5.533    0.000    0.841    0.841
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x4                0.316    0.036    8.867    0.000    0.316    0.511
##    .x6                0.308    0.034    8.939    0.000    0.308    0.448
##    .x7                0.308    0.033    9.282    0.000    0.308    0.425
##    .x8                0.317    0.045    7.088    0.000    0.317    0.392
##    .x9                0.207    0.026    7.966    0.000    0.207    0.395
##    .x24               0.314    0.033    9.551    0.000    0.314    0.397
##    .x26               0.248    0.031    7.978    0.000    0.248    0.392
##    .x30               0.227    0.027    8.316    0.000    0.227    0.370
##    .x31               0.251    0.033    7.516    0.000    0.251    0.397
##    .x14               0.319    0.044    7.280    0.000    0.319    0.461
##    .x19               0.249    0.034    7.368    0.000    0.249    0.335
##    .x20               0.255    0.035    7.211    0.000    0.255    0.366
##    .x21               0.228    0.035    6.571    0.000    0.228    0.330
##    .x25               0.157    0.024    6.499    0.000    0.157    0.251
##    .x1                0.287    0.062    4.619    0.000    0.287    0.484
##    .x2                0.316    0.076    4.131    0.000    0.316    0.506
##    .x27               0.238    0.073    3.267    0.001    0.238    0.378
##    .x28               0.167    0.062    2.682    0.007    0.167    0.282
##    .x5                0.274    0.033    8.214    0.000    0.274    0.523
##    .x10               0.309    0.041    7.465    0.000    0.309    0.477
##    .x16               0.245    0.036    6.728    0.000    0.245    0.473
##    .x17               0.213    0.039    5.520    0.000    0.213    0.312
##    .x18               0.211    0.036    5.871    0.000    0.211    0.328
##    .CM                1.000                               0.171    0.171
```

```
##    .HOTS              1.000                                    0.271    0.271
##    .PAID              1.000                                    0.449    0.449
##    .CC                1.000                                    0.292    0.292
##     TEBS              1.000                                    1.000    1.000
```

```r
# mod_ind <- modificationindices(fit_TEBS)
# head(mod_ind[order(mod_ind$mi, decreasing=TRUE), ], 10)
semPaths(fit_TEBS, "std", title = FALSE)
```




```r
parameterEstimates(fit_TEBS, standardized=TRUE) %>%
  filter(op == "=~") %>%
  select('Latent Factor'=lhs, Indicator=rhs, B=est, SE=se, Z=z, 'p-value'=pvalue, Beta=std.all) %>%
  kable(digits = 3, format="pandoc", caption="Factor Loadings")
```

Table 1: Factor Loadings

Latent Factor	Indicator	B	SE	Z	p-value	Beta
CM	x4	0.227	0.036	6.393	0	0.699
CM	x6	0.254	0.040	6.437	0	0.743
CM	x7	0.267	0.042	6.325	0	0.758
CM	x8	0.290	0.051	5.673	0	0.780
CM	x9	0.233	0.040	5.867	0	0.778
CM	x24	0.285	0.047	6.086	0	0.777
CM	x26	0.256	0.041	6.218	0	0.780
CM	x30	0.257	0.042	6.175	0	0.794
CM	x31	0.255	0.044	5.825	0	0.777
HOTS	x14	0.318	0.037	8.524	0	0.734
HOTS	x19	0.366	0.043	8.475	0	0.816
HOTS	x20	0.346	0.047	7.319	0	0.796

Latent Factor	Indicator	B	SE	Z	p-value	Beta
HOTS	x21	0.354	0.048	7.437	0	0.818
HOTS	x25	0.356	0.047	7.654	0	0.866
PAID	x1	0.371	0.042	8.882	0	0.718
PAID	x2	0.372	0.040	9.208	0	0.703
PAID	x27	0.419	0.097	4.323	0	0.788
PAID	x28	0.437	0.093	4.676	0	0.848
CC	x5	0.270	0.035	7.696	0	0.691
CC	x10	0.315	0.040	7.899	0	0.723
CC	x16	0.282	0.035	8.104	0	0.726
CC	x17	0.370	0.058	6.384	0	0.829
CC	x18	0.356	0.057	6.263	0	0.820
TEBS	CM	2.204	0.416	5.300	0	0.911
TEBS	HOTS	1.641	0.286	5.743	0	0.854
TEBS	PAID	1.107	0.294	3.764	0	0.742
TEBS	CC	1.556	0.281	5.533	0	0.841

SMTI

```
SMTI_model_fi <- '
CA =~ x3 + x4 + x5
CV =~ x6 + x7
SR =~ x10 + x11
SI =~ x13 + x14 + x15 + x16 + x17 + x18
'
SMTI_model_se <- '
CA =~ x3 + x4 + x5
CV =~ x6 + x7
SR =~ x10 + x11
SI =~ x13 + x14 + x15 + x16 + x17 + x18
Identity =~ CA + CV + SR + SI'
```

```
fit_SMTI<- cfa(SMTI_model_se,
               data=Data_SMTI,estimator = "MLR", std.lv = TRUE)
```

```
## Warning in lav_object_post_check(object): lavaan WARNING: some estimated ov
## variances are negative
```

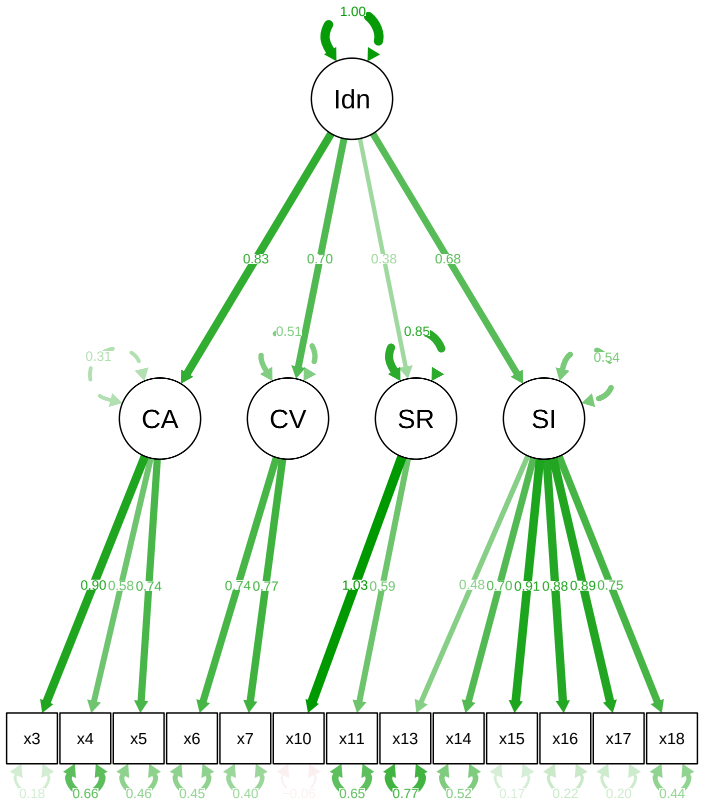
```
summary(fit_SMTI, standardized = TRUE,
        fit.measures = TRUE)
```

```
## lavaan 0.6-5 ended normally after 57 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         30
##
##   Number of observations                           165
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               135.539     107.111
##   Degrees of freedom                                61          61
```

```
##   P-value (Chi-square)                          0.000       0.000
##   Scaling correction factor                                 1.265
##     for the Yuan-Bentler correction (Mplus variant)
##
## Model Test Baseline Model:
##
##   Test statistic                             1185.203     880.877
##   Degrees of freedom                               78          78
##   P-value                                       0.000       0.000
##   Scaling correction factor                                 1.345
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                   0.933       0.943
##   Tucker-Lewis Index (TLI)                      0.914       0.927
##
##   Robust Comparative Fit Index (CFI)                        0.946
##   Robust Tucker-Lewis Index (TLI)                           0.931
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)             -1785.500   -1785.500
##   Scaling correction factor                                 1.706
##       for the MLR correction
##   Loglikelihood unrestricted model (H1)     -1717.731   -1717.731
##   Scaling correction factor                                 1.411
##       for the MLR correction
##
##   Akaike (AIC)                               3631.000    3631.000
##   Bayesian (BIC)                             3724.178    3724.178
##   Sample-size adjusted Bayesian (BIC)        3629.198    3629.198
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                         0.086       0.068
##   90 Percent confidence interval - lower        0.067       0.048
##   90 Percent confidence interval - upper        0.106       0.086
##   P-value RMSEA <= 0.05                         0.002       0.064
##
##   Robust RMSEA                                              0.076
##   90 Percent confidence interval - lower                    0.052
##   90 Percent confidence interval - upper                    0.100
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                          0.062       0.062
##
## Parameter Estimates:
##
##   Information                                      Observed
##   Observed information based on                     Hessian
##   Standard errors                         Robust.huber.white
##
## Latent Variables:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   CA =~
##     x3                 0.422    0.113    3.737    0.000    0.757    0.904
##     x4                 0.340    0.084    4.042    0.000    0.610    0.581
##     x5                 0.303    0.089    3.419    0.001    0.543    0.738
##   CV =~
##     x6                 0.285    0.051    5.564    0.000    0.400    0.741
##     x7                 0.272    0.045    6.065    0.000    0.382    0.773
##   SR =~
##     x10                0.633    0.132    4.797    0.000    0.685    1.029
##     x11                0.385    0.073    5.276    0.000    0.416    0.591
##   SI =~
##     x13                0.297    0.057    5.175    0.000    0.405    0.485
##     x14                0.353    0.063    5.630    0.000    0.481    0.696
##     x15                0.484    0.074    6.513    0.000    0.659    0.912
##     x16                0.461    0.067    6.916    0.000    0.628    0.884
##     x17                0.497    0.076    6.526    0.000    0.678    0.894
##     x18                0.352    0.060    5.839    0.000    0.480    0.746
##   Identity =~
##     CA                 1.489    0.485    3.070    0.002    0.830    0.830
##     CV                 0.986    0.309    3.189    0.001    0.702    0.702
##     SR                 0.414    0.188    2.199    0.028    0.382    0.382
##     SI                 0.927    0.255    3.629    0.000    0.680    0.680
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .x3                 0.129    0.054    2.376    0.017    0.129    0.184
##    .x4                 0.733    0.094    7.819    0.000    0.733    0.663
##    .x5                 0.246    0.046    5.312    0.000    0.246    0.455
##    .x6                 0.131    0.033    4.023    0.000    0.131    0.450
##    .x7                 0.098    0.030    3.228    0.001    0.098    0.403
##    .x10               -0.026    0.156   -0.170    0.865   -0.026   -0.060
##    .x11                0.322    0.067    4.778    0.000    0.322    0.650
##    .x13                0.534    0.066    8.106    0.000    0.534    0.765
##    .x14                0.247    0.042    5.840    0.000    0.247    0.516
##    .x15                0.088    0.021    4.230    0.000    0.088    0.169
##    .x16                0.110    0.036    3.090    0.002    0.110    0.219
##    .x17                0.116    0.023    5.118    0.000    0.116    0.202
##    .x18                0.184    0.032    5.740    0.000    0.184    0.444
##    .CA                 1.000                               0.311    0.311
##    .CV                 1.000                               0.507    0.507
##    .SR                 1.000                               0.854    0.854
##    .SI                 1.000                               0.538    0.538
##     Identity           1.000                               1.000    1.000
```

```r
# mod_ind <- modificationindices(fit_SMTI)
# head(mod_ind[order(mod_ind$mi, decreasing=TRUE), ], 10)
semPaths(fit_SMTI, "std", title = FALSE)
```




```r
parameterEstimates(fit_SMTI, standardized=TRUE) %>%
  filter(op == "=~") %>%
  select('Latent Factor'=lhs, Indicator=rhs, B=est, SE=se, Z=z, 'p-value'=pvalue, Beta=std.all) %>%
  kable(digits = 3, format="pandoc", caption="Factor Loadings")
```

Table 2: Factor Loadings

Latent Factor	Indicator	B	SE	Z	p-value	Beta
CA	x3	0.422	0.113	3.737	0.000	0.904
CA	x4	0.340	0.084	4.042	0.000	0.581
CA	x5	0.303	0.089	3.419	0.001	0.738
CV	x6	0.285	0.051	5.564	0.000	0.741
CV	x7	0.272	0.045	6.065	0.000	0.773
SR	x10	0.633	0.132	4.797	0.000	1.029
SR	x11	0.385	0.073	5.276	0.000	0.591
SI	x13	0.297	0.057	5.175	0.000	0.485
SI	x14	0.353	0.063	5.630	0.000	0.696
SI	x15	0.484	0.074	6.513	0.000	0.912
SI	x16	0.461	0.067	6.916	0.000	0.884
SI	x17	0.497	0.076	6.526	0.000	0.894
SI	x18	0.352	0.060	5.839	0.000	0.746
Identity	CA	1.489	0.485	3.070	0.002	0.830
Identity	CV	0.986	0.309	3.189	0.001	0.702
Identity	SR	0.414	0.188	2.199	0.028	0.382
Identity	SI	0.927	0.255	3.629	0.000	0.680

6.2 Network Explanation: Energizing and Frequency

First, we need to load detailed network of each teacher.

```
load(file = "~/20200614_EFA_Data_network.RData")
Ecolor = rev(gray.colors(5))
```

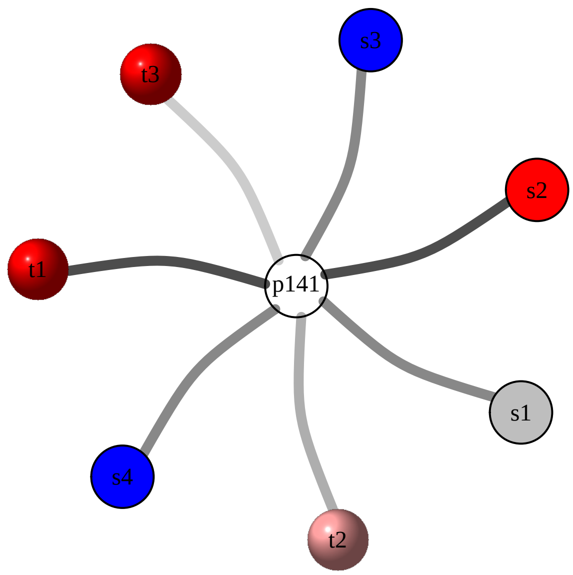
Second, we use *igraph* package to build the function in order to plot the individual social network.

```
network_plot <- function(id){
  plot_test <- network_details[network_details$id == id,]

  ## plots ##
  edges <- plot_test
  nodes <- plot_test[c("contact")]
  nodes$id <- 1:nrow(plot_test)
  de <- data.frame(id,nrow(plot_test)+1)
  names(de) <- c("contact","id")
  nodes <- rbind(nodes, de)

  g <- graph_from_data_frame(d=edges, vertices=nodes, directed=FALSE)
  op_side_5 <- plot_test$contact[plot_test$energize == 5]
  op_side_4 <- plot_test$contact[plot_test$energize == 4]
  ne_side <- plot_test$contact[plot_test$energize == 3]
  de_side_2 <- plot_test$contact[plot_test$energize == 2]
  de_side_1 <- plot_test$contact[plot_test$energize == 1]
  V(g)$color <- NA
  V(g)$color[V(g)$name %in% op_side_5] <- "red"
  V(g)$color[V(g)$name %in% op_side_4] <- "#FFA2A2"
  V(g)$color[V(g)$name %in% ne_side] <- "grey"
  V(g)$color[V(g)$name %in% de_side_2] <- "#A2A2FF"
  V(g)$color[V(g)$name %in% de_side_1] <- "blue"
  # vertex_attr(g)

  school <- plot_test$contact[startsWith(plot_test$contact,'s')]
  district <- plot_test$contact[startsWith(plot_test$contact,'d')]
  state <- plot_test$contact[startsWith(plot_test$contact,'t')]
  nation <- plot_test$contact[startsWith(plot_test$contact,'n')]
  V(g)$shape <- "circle"
  V(g)$shape[V(g)$name %in% school] <- "circle"
  V(g)$shape[V(g)$name %in% district] <- "square"
  V(g)$shape[V(g)$name %in% state] <- "sphere"
  V(g)$shape[V(g)$name %in% nation] <- "crectangle"

  op_side_5 <- plot_test$contact[plot_test$energize == 5]
  op_side_4 <- plot_test$contact[plot_test$energize == 4]
  ne_side <- plot_test$contact[plot_test$energize == 3]
  de_side_2 <- plot_test$contact[plot_test$energize == 2]
  de_side_1 <- plot_test$contact[plot_test$energize == 1]

  E(g)$color <- Ecolor[E(g)$freq]
  E(g)$width <- 5
  # edge_attr(g)

  V(g)$size <- 25
```

```
  plot(g,
       vertex.label.color = "black",
       vertex.label.cex = .75,
       edge.curved=.25
  )
}
```

As a example, here is the network plot of teacher “p141”

```
network_plot("p141")
```



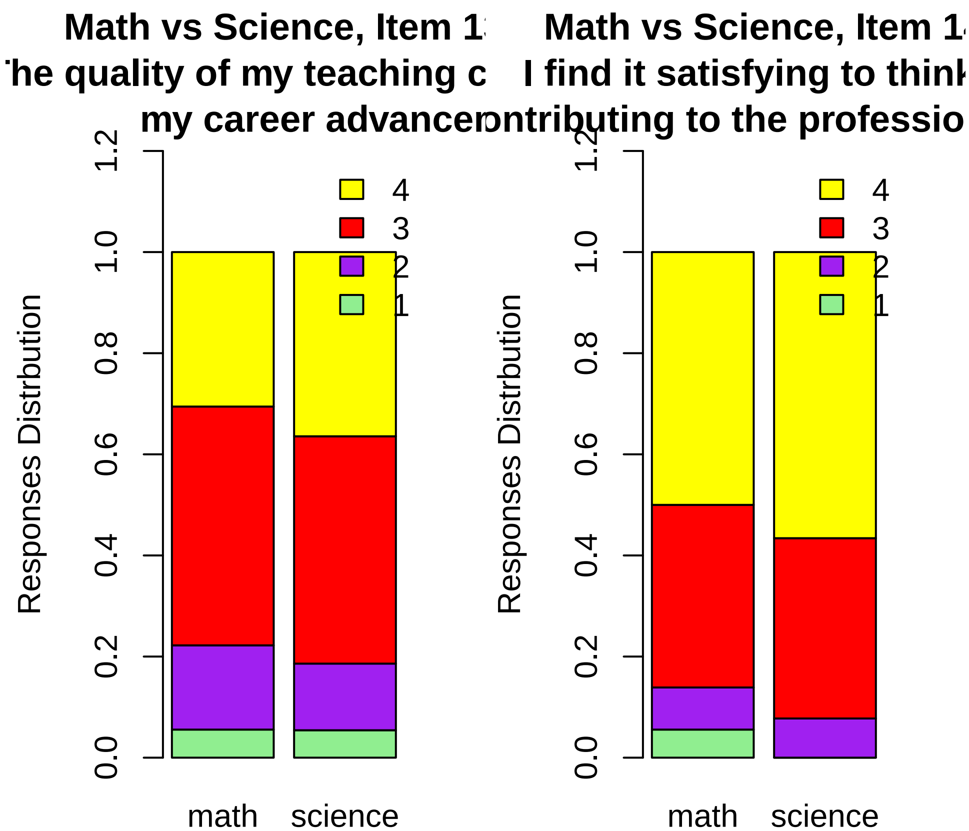

```
corrdata[corrdata$id =="p141", ]$avgfreq_overall
```

```
## [1] 3.857143
```

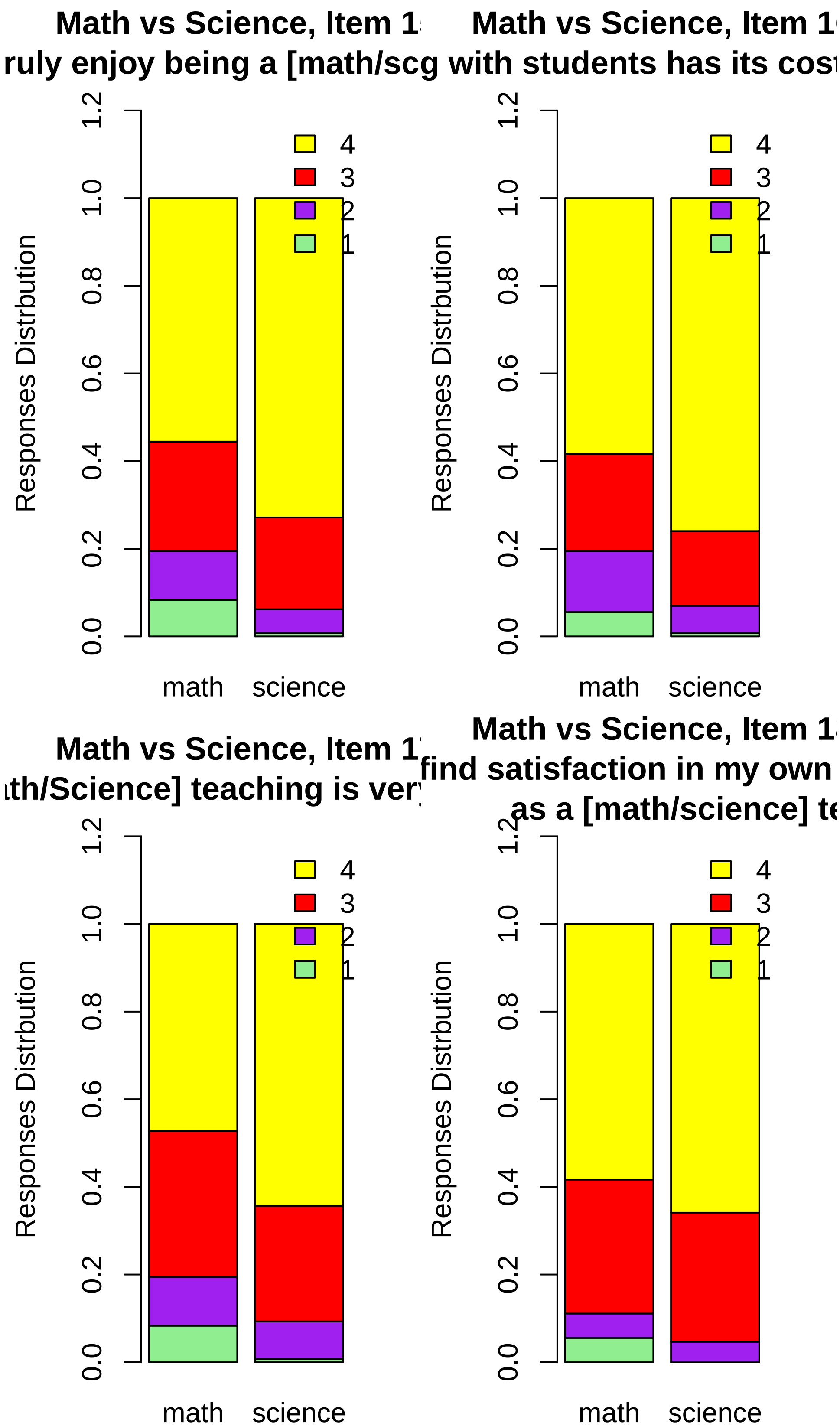
```
corrdata[corrdata$id =="p141", ]$avgenergize_overall
```

```
## [1] 3.428571
```

6.3 SI items distribution plots



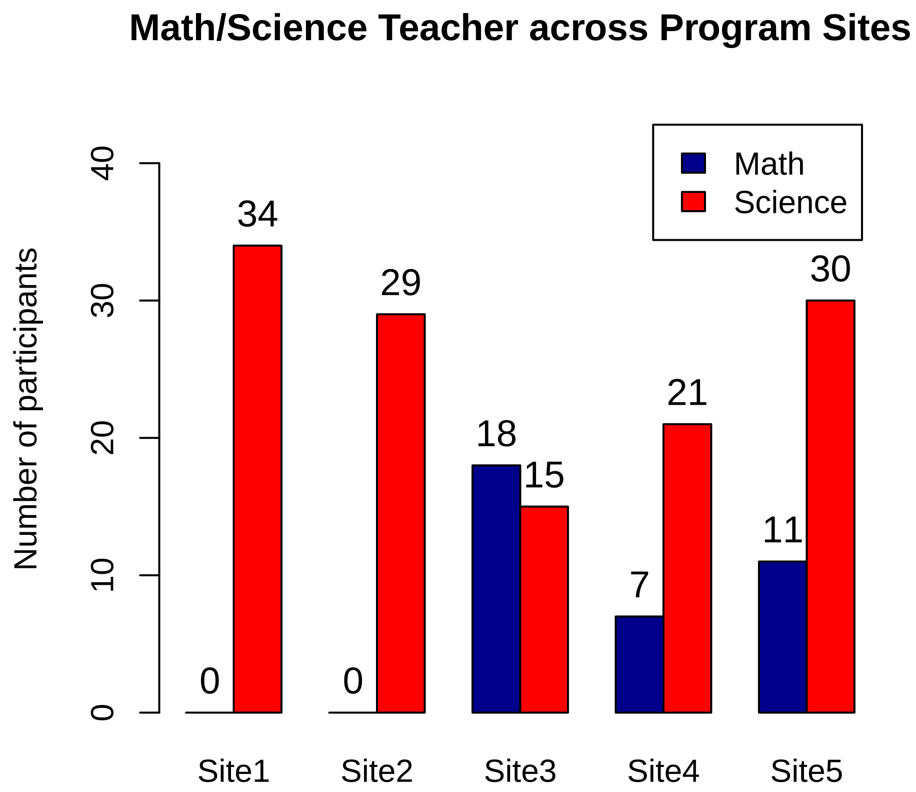

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on ' Working with students has its costs, but it's worth
## it.' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on ' Working with students has its costs, but it's worth
## it.' in 'mbcsToSbcs': dot substituted for <80>

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## conversion failure on ' Working with students has its costs, but it's worth
## it.' in 'mbcsToSbcs': dot substituted for <98>
```


Math vs Science, Item 1
ruly enjoy being a [math/scg
Math vs Science, Item 1
with students has its cost
Math vs Science, Item 1
ıth/Science] teaching is ver
Math vs Science, Item 1
find satisfaction in my own
as a [math/science] te
Responses Distrbution
0.0
0.2
0.4
0.6
0.8
1.0
1.2
4
3
2
1
math
science

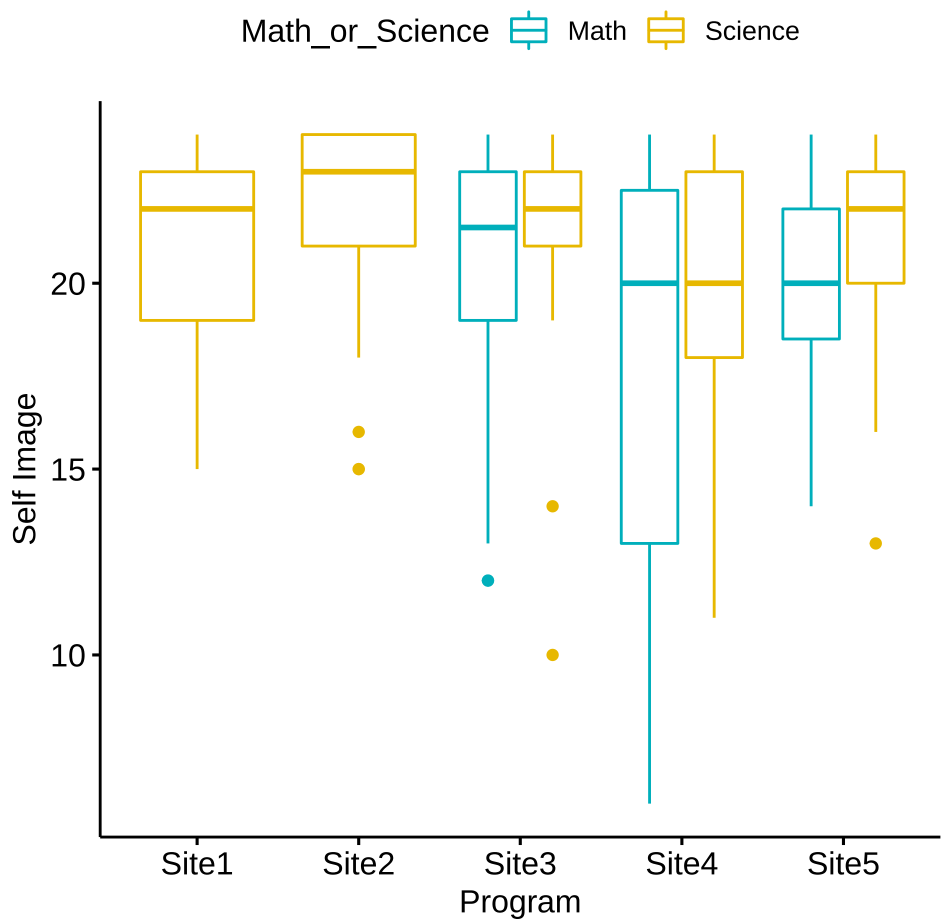
6.4 Moderation analysis of program site effect on subject related Self-Image scores.

```
program <- data[c("id","pg1","Math_or_Science","SI")]
program$Math_or_Science <- as.factor(program$Math_or_Science)
str(program)
levels(program$Math_or_Science) <- c("Math","Science")
```

```
counts = table(program$Math_or_Science, program$pg1)
ylim <- c(0, 1.3*max(counts))
par(mfrow=c(1,1))
xx <- barplot(counts, main="Math/Science Teacher across Program Sites",
              col=c("darkblue","red"),
              legend = rownames(counts),
              beside=TRUE,ylim = ylim,width=0.85,xlab='',ylab='Number of participants'
              #             ,args.legend = list(x = "right", inset=c(-0.35, 0))
)
text(x = xx, y = counts, label = counts, pos = 3, cex = 1.2, col = "black")
```




```
test <- program[c("id","pg1","SI","Math_or_Science")]
names(test)[2] <- "program"
ggboxplot(test, x = "program", y = "SI", color = "Math_or_Science", xlab = "Program", ylab = "Self Image
          palette = c("#00AFBB", "#E7B800"))
```



ANOVA Table

```
SI.aov <- aov(SI ~ Math_or_Science + program, data = test)
summary(SI.aov) # 0.019 *  0.163
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## Math_or_Science   1   67.9   67.88   5.618  0.019 *
## program           4   80.1   20.02   1.657  0.163
## Residuals       159 1921.3   12.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(SI.aov)
```

```
## Analysis of Variance Table
##
## Response: SI
##                  Df  Sum Sq Mean Sq F value  Pr(>F)
## Math_or_Science   1   67.88  67.879  5.6175 0.01898 *
## program           4   80.10  20.024  1.6572 0.16261
## Residuals       159 1921.27  12.083
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
SI.aov3 <- aov(SI ~ Math_or_Science * program, data = test)
summary(SI.aov3) # 0.0193 * 0.1647
```

```
##                           Df Sum Sq Mean Sq F value Pr(>F)
## Math_or_Science            1   67.9   67.88   5.590 0.0193 *
## program                    4   80.1   20.02   1.649 0.1647
## Math_or_Science:program    2   14.7    7.37   0.607 0.5464
## Residuals                157 1906.5   12.14
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(SI.aov3)
```

```
## Analysis of Variance Table
##
## Response: SI
##                           Df  Sum Sq Mean Sq F value  Pr(>F)
## Math_or_Science            1   67.88  67.879  5.5897 0.01929 *
## program                    4   80.10  20.024  1.6490 0.16466
## Math_or_Science:program    2   14.74   7.368  0.6068 0.54639
## Residuals                157 1906.54  12.144
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```