论文数据分析

郑元瑞

目录

1 加载包和数据预处理

清除环境中其他变量

```
rm(list = ls())
```

1.1 加载包

```
library(tidyverse)
library(report)
library(sandwich)
library(readxl)
library(psych)
library(writexl)
library(lavaan)
library(tidySEM)
library(sjmisc)
library(Hmisc)
library(performance)
library(rockchalk)
library(interactions)
library(semTools)
library(effectsize)
# install.packages("lavaanPlot")
library(lavaanPlot)
```

1.2 数据预处理

```
df <- read_excel(
  "/Users/zhengyuanrui/Desktop/社交网站/数据/原始数据.xlsx"
)
colnames(df) <-c("id","times_submit",
                    "time_finish",
                    "source","source_info","IP",
                    "sex","grade","residence",
                    "school","SNS1","SNS2",
                    "SNS3","SNS4","SNS5","SNS6",
                    "SES1","SES2","SES3","SES4","SES5",
                    "SES6","SES7","SES8","SES9",
                    "SES10","CSES1","CSES2","CSES3",
                    "CSES4","EIB1","EIB2","EIB3","EIB4",
                    "EIB5","EIB6","EIB7","EIB8","total")
df <- df %>%
  rec(SES3, rec = "1 = 4; 2 = 3; 3 = 2; 4 = 1") %>%
  #
  rec(SES5, rec = "1 = 4; 2 = 3; 3 = 2; 4 = 1") %>%
  rec(SES8, rec = "1 = 4; 2 = 3; 3 = 2; 4 = 1") %>%
  rec(SES9, rec = "1 = 4; 2 = 3; 3 = 2; 4 = 1") %>%
  rec(SES10, rec = "1 = 4; 2 = 3; 3 = 2; 4 = 1")

df1 <- df
df1$SNS1 <- scale(df1$SNS1, center = T, scale = T)
df1$SNS2 <- scale(df1$SNS2, center = T, scale = T)
df1$SNS3 <- scale(df1$SNS3, center = T, scale = T)
df1$SNS4 <- scale(df1$SNS4, center = T, scale = T)
df1$SNS5 <- scale(df1$SNS5, center = T, scale = T)
df1$SNS6 <- scale(df1$SNS6, center = T, scale = T)
# write_xlsx(df1, " 信效度.xlsx")
# getwd()
```

2 共同方法偏差检验

2.1 harman

```
h <- df %>% dplyr::select(-c(id, times_submit,
                      time_finish, source,
                      source_info, IP, sex,
                      grade, residence, school,
                      total,SES3, SES5, SES9, SES10, SES8_r))
h <- as.data.frame(h)
# write_csv(h, "harman.csv")

singlefactor <- psych::fa(h, nfactors = 1, rotate = "none", fm="ml")

singlefactor[["Vaccounted"]]
```

```
##                       ML1
## SS loadings     9.8578723
## Proportion Var 0.3520669
```

2.2 双因子

```
model1 <- '
  # measurement model
    SNS =~ SNS1 + SNS2 + SNS3 + SNS4 +
    SNS5 + SNS6
    SES =~ SES1 + SES2 + SES3_r + SES4 +
    SES5_r + SES6 + SES7 + SES8 + SES9_r + SES10_r
    CSES =~ CSES1 + CSES2 + CSES3 + CSES4
    EIB =~ EIB1 + EIB2 + EIB3 + EIB4 +
    EIB5 + EIB6 + EIB7 + EIB8
  # residual correlations
```

```
  SNS ~~ SES
  SNS ~~ CSES
  SNS ~~ EIB
  SES ~~ CSES
  SES ~~ EIB
  CSES ~~ EIB
'
model2 <- '
  # measurement model
    SNS =~ SNS1 + SNS2 + SNS3 + SNS4 + SNS5 +
    SNS6
    SES =~ SES1 + SES2 + SES3_r + SES4 + SES5_r +
    SES6 + SES7 + SES8 + SES9_r + SES10_r

    CSES =~ CSES1 + CSES2 + CSES3 + CSES4
    EIB =~ EIB1 + EIB2 + EIB3 + EIB4 + EIB5 +
    EIB6 + EIB7 + EIB8

    f =~ 1*SNS1 + 1*SNS2 + 1*SNS3 + 1*SNS4 +
    1*SNS5 + 1*SNS6+1*SES1 + 1*SES2 +
    1*SES3_r + 1*SES4 + 1*SES5_r + 1*SES6 +
    1*SES7 + 1*SES8 + 1*SES9_r + 1*SES10_r+
    1*CSES1 + 1*CSES2 + 1*CSES3 + 1*CSES4 +
    1*EIB1 +
    1*EIB2 + 1*EIB3 + 1*EIB4 + 1*EIB5 +
    1*EIB6 + 1*EIB7 + 1*EIB8

  # residual correlations
  SNS ~~ SES
  SNS ~~ CSES
  SNS ~~ EIB
  SES ~~ CSES
  SES ~~ EIB
```

```
  CSES ~~ EIB
'
fit <- cfa(model1, data = df)
fit2 <- cfa(model2, data = df)
```

```
fitMeasures(fit, c("chisq", "df",
                   "pvalue","cfi", "rmsea", "tli", 'srmr'))
```

```
##    chisq       df   pvalue      cfi    rmsea      tli     srmr
## 4029.410  344.000    0.000    0.806    0.103    0.787    0.098
```

```
fitMeasures(fit2, c("chisq", "df",
                   "pvalue","cfi", "rmsea", "tli", 'srmr'))
```

```
##    chisq       df   pvalue      cfi    rmsea      tli     srmr
## 2393.989  339.000    0.000    0.892    0.077    0.879    0.087
```

3 问卷信效度分析

3.1 社交网站使用强度

3.1.1 信效度分析

```
valid_SNS <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6"
model_valid_SNS <- cfa(valid_SNS, data = df1)
fitMeasures(model_valid_SNS,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 361.249   9.000   0.000   0.889   0.196   0.816   0.065
```

```
df1 %>% dplyr::select(SNS1:SNS6) %>%
  KMO()
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = .)
## Overall MSA =  0.86
## MSA for each item =
## SNS1 SNS2 SNS3 SNS4 SNS5 SNS6
## 0.83 0.88 0.86 0.87 0.87 0.86
```

```
df1 %>% filter(school == 1) -> df_public
cfa(valid_SNS, data = df_public) %>%
  fitMeasures(c("chisq", "df", "pvalue",
                "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 168.102   9.000   0.000   0.884   0.206   0.807   0.067
```

```
df1 %>% filter(school == 2) -> df_private
cfa(valid_SNS, data = df_private) %>%
  fitMeasures(c("chisq", "df", "pvalue",
                "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 209.051   9.000   0.000   0.886   0.193   0.810   0.069
```

```
valid_SNS <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6"
model_valid_SNS <- cfa(valid_SNS, data = df1)
modindices(model_valid_SNS, sort = TRUE, maximum.number = 5)
```

3.1.1.1 模型修正

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 15 SNS1 ~~ SNS3 88.760  0.160   0.160    0.431    0.431
```

```
## 18 SNS1 ~~ SNS6 88.441 -0.174  -0.174   -0.354   -0.354
## 14 SNS1 ~~ SNS2 87.438  0.167   0.167    0.365    0.365
## 28 SNS5 ~~ SNS6 85.554  0.169   0.169    0.354    0.354
## 27 SNS4 ~~ SNS6 77.396  0.164   0.164    0.330    0.330
```

```
fitMeasures(model_valid_SNS,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 361.249   9.000   0.000   0.889   0.196   0.816   0.065
```

第一次修正

```
valid_SNS2 <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6
SNS1    ~~  SNS3
"
model_valid_SNS2 <- cfa(valid_SNS2, data = df1)
fitMeasures(model_valid_SNS2,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 263.109   8.000   0.000   0.920   0.177   0.850   0.063
```

```
modindices(model_valid_SNS2, sort = TRUE, maximum.number = 5)
```

```
##     lhs op  rhs      mi    epc sepc.lv sepc.all sepc.nox
## 15 SNS1 ~~ SNS2 105.430  0.181   0.181    0.329    0.329
## 20 SNS2 ~~ SNS4  47.205 -0.130  -0.130   -0.285   -0.285
## 18 SNS1 ~~ SNS6  37.394 -0.105  -0.105   -0.201   -0.201
## 28 SNS5 ~~ SNS6  24.788  0.093   0.093    0.228    0.228
## 27 SNS4 ~~ SNS6  24.353  0.092   0.092    0.214    0.214
```

第二次修正

```
valid_SNS3 <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6
SNS1    ~~  SNS3
SNS1    ~~  SNS2
"
model_valid_SNS3 <- cfa(valid_SNS3, data = df1)
fitMeasures(model_valid_SNS3,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 157.982   7.000   0.000   0.953   0.146   0.898   0.046
```

```
modindices(model_valid_SNS3, sort = TRUE, maximum.number = 5)
```

```
##     lhs op  rhs      mi    epc sepc.lv sepc.all sepc.nox
## 19 SNS2 ~~ SNS3 116.400  0.212   0.212    0.419    0.419
## 24 SNS3 ~~ SNS5  29.241 -0.090  -0.090   -0.237   -0.237
## 20 SNS2 ~~ SNS4  27.107 -0.092  -0.092   -0.200   -0.200
## 18 SNS1 ~~ SNS6  17.820 -0.069  -0.069   -0.133   -0.133
## 28 SNS5 ~~ SNS6  16.060  0.076   0.076    0.190    0.190
```

第三次修正

```
valid_SNS4 <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6
SNS1    ~~  SNS3
SNS1    ~~  SNS2
SNS2    ~~  SNS3
"
model_valid_SNS4 <- cfa(valid_SNS4, data = df1)
fitMeasures(model_valid_SNS4,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##  chisq     df pvalue    cfi  rmsea    tli   srmr
```

```
## 47.634  6.000  0.000  0.987  0.083  0.967  0.022
```

```
modindices(model_valid_SNS4, sort = TRUE, maximum.number = 5)
```

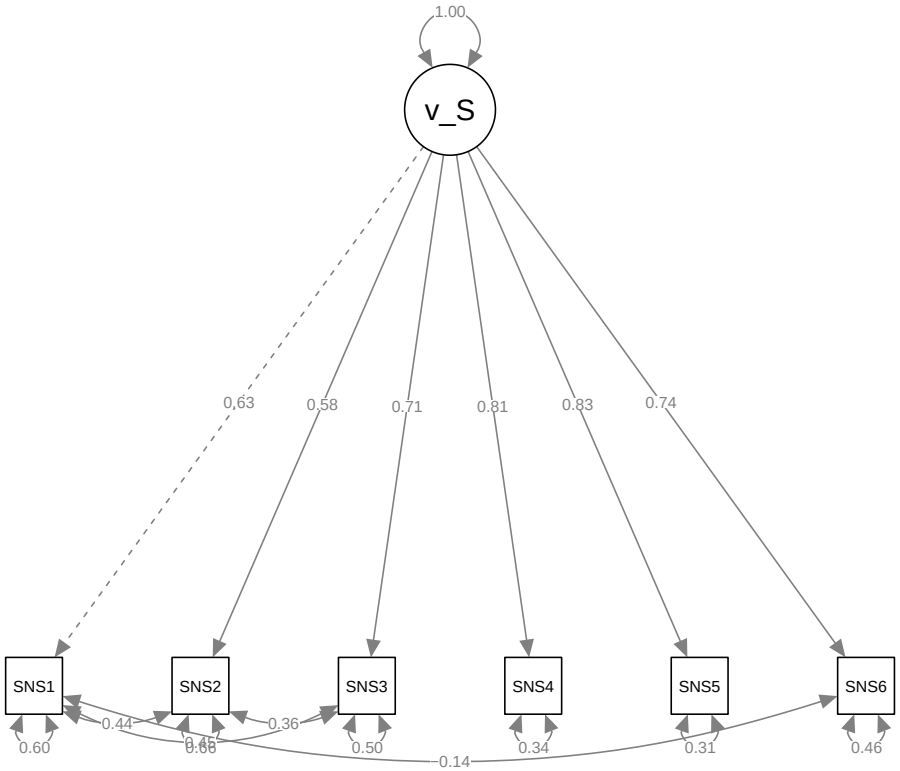
```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 19 SNS1 ~~ SNS6 18.645 -0.069  -0.069   -0.127   -0.127
## 23 SNS3 ~~ SNS4 13.592  0.056   0.056    0.138    0.138
## 18 SNS1 ~~ SNS5 13.404  0.055   0.055    0.125    0.125
## 26 SNS4 ~~ SNS5 12.707 -0.088  -0.088   -0.272   -0.272
## 24 SNS3 ~~ SNS5 10.792 -0.050  -0.050   -0.128   -0.128
```

第四次修正

```
valid_SNS5 <- "v_SNS=~ SNS1 + SNS2 +
SNS3 + SNS4 + SNS5 + SNS6
SNS1    ~~  SNS3
SNS1    ~~  SNS2
SNS2    ~~  SNS3
SNS1    ~~  SNS6
"
model_valid_SNS5 <- cfa(valid_SNS5, data = df1)
fitMeasures(model_valid_SNS5,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##  chisq     df pvalue    cfi  rmsea    tli   srmr
## 27.692  5.000  0.000  0.993  0.067  0.979  0.015
```

```
library(semPlot)
semPaths(object = model_valid_SNS5, whatLabels = "std",)
```




```
semTools::reliability(model_valid_SNS)
```

```
##            v_SNS
## alpha  0.8822714
## omega  0.8831967
## omega2 0.8831967
## omega3 0.8834165
## avevar 0.5587465
```

3.2 自尊

3.2.1 信效度分析

```
valid_SES <- "v_SES =~ SES1 + SES2 +
SES3_r + SES4 + SES5_r + SES6 + SES7 +
```

```
SES8 + SES9_r + SES10_r
"
model_valid_SES <- cfa(valid_SES, data = df1)
fitMeasures(model_valid_SES,
            c("chisq", "df", "pvalue",
              "cfi", "rmsea", "tli", 'srmr'))
```

```
##    chisq       df   pvalue      cfi    rmsea      tli     srmr
## 2208.133   35.000    0.000    0.555    0.247    0.428    0.204
```

```
df1 %>% dplyr::select(SES1, SES2, SES3_r,
      SES4, SES5_r, SES6, SES7,
      SES8, SES9_r, SES10_r) %>% KMO()
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = .)
## Overall MSA =  0.83
## MSA for each item =
##    SES1    SES2  SES3_r    SES4  SES5_r    SES6    SES7    SES8  SES9_r SES10_r
##    0.88    0.87    0.81    0.89    0.80    0.84    0.85    0.89    0.74    0.74
```

```
df1 %>% dplyr::select(SES1, SES2, SES3_r,
      SES4, SES5_r, SES6, SES7,
      SES8, SES9_r, SES10_r) %>%
  cortest.bartlett()
```

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

```
## $chisq
## [1] 4901.013
##
## $p.value
## [1] 0
##
## $df
## [1] 45
```

```
df1 %>% dplyr::select(SES1, SES2, SES3_r,
      SES4, SES5_r, SES6, SES7,
      SES8, SES9_r, SES10_r) %>%
  principal(nfactors=1, score=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = ., nfactors = 1, scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##           PC1     h2   u2 com
## SES1     0.69 0.4744 0.53   1
## SES2     0.77 0.5893 0.41   1
## SES3_r   0.53 0.2799 0.72   1
## SES4     0.70 0.4936 0.51   1
## SES5_r   0.49 0.2440 0.76   1
## SES6     0.77 0.5989 0.40   1
## SES7     0.75 0.5634 0.44   1
## SES8    -0.10 0.0099 0.99   1
## SES9_r   0.51 0.2594 0.74   1
## SES10_r  0.51 0.2642 0.74   1
##
##                 PC1
## SS loadings    3.78
## Proportion Var 0.38
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.23
##  with the empirical chi square  4779.38  with prob <  0
##
## Fit based upon off diagonal values = 0.64
```

```
semTools::reliability(model_valid_SES)
```

```
##            v_SES
## alpha  0.7498921
## omega  0.7215033
## omega2 0.7215033
## omega3 0.6300458
## avevar 0.2519320
```

```
M8 <- "
PSES =~ SES1 + SES2 + SES4 + SES6 + SES7

NSES =~ SES3_r + SES5_r + SES8 + SES9_r + SES10_r

GSES =~ SES1 + SES2 + SES3_r + SES4 + SES5_r + SES6 + SES7 + SES8 + SES9_r + SES10_r

PSES ~~ NSES
GSES ~~ 0*NSES
GSES ~~ 0*PSES
"

M8_cfa <- cfa(M8,df1,optim.method="BFGS",optim.force.converged=T,check.post=F)
fitMeasures(M8_cfa, c("chisq", "df", "pvalue","cfi", "rmsea", "tli", 'srmr'))
```

3.2.1.1 自尊量表双因子模型

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 150.686  24.000   0.000   0.974   0.072   0.951   0.047
```

3.3 创新自我效能感

3.3.1 信效度分析

```
valid_CSES <- "v_CSES =~ CSES1 + CSES2 +
CSES3 + CSES4"
model_valid_CSES <- cfa(valid_CSES, data = df1)
fitMeasures(model_valid_CSES,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", 'srmr'))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr
## 122.206   2.000   0.000   0.962   0.243   0.885   0.030
```

```
modindices(model_valid_CSES, sort = TRUE, maximum.number = 5)
```

```
##      lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
## 15 CSES3 ~~ CSES4 99.505  0.218   0.218    0.410    0.410
## 10 CSES1 ~~ CSES2 99.505  0.286   0.286    0.883    0.883
## 13 CSES2 ~~ CSES3 75.734 -0.199  -0.199   -0.494   -0.494
## 12 CSES1 ~~ CSES4 75.734 -0.209  -0.209   -0.489   -0.489
## 14 CSES2 ~~ CSES4  0.428 -0.017  -0.017   -0.051   -0.051
```

3.3.1.1 模型修正 一次修正

```
valid_CSES2 <- "v_CSES =~ CSES1 + CSES2 +
CSES3 + CSES4
CSES3   ~~  CSES4
"
model_valid_CSES2 <- cfa(valid_CSES2, data = df1)
fitMeasures(model_valid_CSES2,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", 'srmr'))
```

```
##  chisq     df pvalue    cfi  rmsea    tli   srmr
## 26.427  1.000  0.000  0.992  0.158  0.951  0.012
```

```
modindices(model_valid_CSES2, sort = TRUE, maximum.number = 5)
```

```
##      lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
## 14 CSES2 ~~ CSES3 26.086 -0.113  -0.113   -0.310   -0.310
## 15 CSES2 ~~ CSES4 26.086  0.127   0.127    0.417    0.417
## 12 CSES1 ~~ CSES3 26.086  0.106   0.106    0.191    0.191
## 13 CSES1 ~~ CSES4 26.086 -0.119  -0.119   -0.257   -0.257
```

二次修正

```
valid_CSES3 <- "v_CSES =~ CSES1 + CSES2 +
CSES3 + CSES4
CSES3   ~~  CSES4
CSES2   ~~  CSES3
"
model_valid_CSES3 <- cfa(valid_CSES2, data = df1)
fitMeasures(model_valid_CSES3,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", 'srmr'))
```

```
##  chisq     df pvalue    cfi  rmsea    tli   srmr
## 26.427  1.000  0.000  0.992  0.158  0.951  0.012
```

```
modindices(model_valid_CSES3, sort = TRUE, maximum.number = 5)
```

```
##      lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
## 14 CSES2 ~~ CSES3 26.086 -0.113  -0.113   -0.310   -0.310
## 15 CSES2 ~~ CSES4 26.086  0.127   0.127    0.417    0.417
## 12 CSES1 ~~ CSES3 26.086  0.106   0.106    0.191    0.191
## 13 CSES1 ~~ CSES4 26.086 -0.119  -0.119   -0.257   -0.257
```

```
df1 %>% dplyr::select(CSES1:CSES4) %>%
  KMO()
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = .)
## Overall MSA =  0.81
```

```
## MSA for each item =
## CSES1 CSES2 CSES3 CSES4
##  0.80  0.77  0.86  0.82
```

```
df1 %>% dplyr::select(CSES1:CSES4) %>%
  cortest.bartlett()
```

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

## $chisq
## [1] 3122.308
##
## $p.value
## [1] 0
##
## $df
## [1] 6
```

```
df1 %>% select(CSES1:CSES4) %>%
  principal(nfactors=1, score=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = ., nfactors = 1, scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##        PC1   h2   u2 com
## CSES1 0.90 0.81 0.19   1
## CSES2 0.92 0.85 0.15   1
## CSES3 0.86 0.74 0.26   1
## CSES4 0.90 0.82 0.18   1
##
##                 PC1
## SS loadings    3.22
## Proportion Var 0.81
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
```

```
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  71.52  with prob <  2.9e-16
##
## Fit based upon off diagonal values = 0.99
```

```
semTools::reliability(model_valid_CSES)
```

```
##           v_CSES
## alpha  0.9191806
## omega  0.9198976
## omega2 0.9198976
## omega3 0.9176677
## avevar 0.7427416
```

3.4 创新行为

3.4.1 信效度分析

```
valid_EIB <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +
      EIB6 + EIB7 + EIB8"
model_valid_EIB <- cfa(valid_EIB,
                       data = df1)
fitMeasures(model_valid_EIB,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr     gfi
## 446.153  20.000   0.000   0.921   0.145   0.890   0.050   0.878
```

```
modindices(model_valid_EIB, sort = TRUE, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 41 EIB5 ~~ EIB7 85.186  0.141   0.141    0.331    0.331
```

```
## 18 EIB1 ~~ EIB2 84.068  0.094   0.094    0.349    0.349
## 22 EIB1 ~~ EIB6 64.056 -0.094  -0.094   -0.295   -0.295
## 19 EIB1 ~~ EIB3 57.799  0.080   0.080    0.278    0.278
## 43 EIB6 ~~ EIB7 54.929  0.105   0.105    0.269    0.269
```

3.4.1.1 模型修正 一次修正

```
valid_EIB2 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
EIB5    ~~  EIB7
"
model_valid_EIB2 <- cfa(valid_EIB2,
                        data = df1)
fitMeasures(model_valid_EIB2,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr     gfi
## 360.375  19.000   0.000   0.937   0.133   0.907   0.047   0.905
```

```
modindices(model_valid_EIB2, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 19 EIB1 ~~ EIB2 64.177  0.081   0.081    0.316    0.316
## 45 EIB7 ~~ EIB8 63.115  0.099   0.099    0.270    0.270
## 23 EIB1 ~~ EIB6 63.059 -0.095  -0.095   -0.297   -0.297
## 43 EIB6 ~~ EIB7 44.845  0.092   0.092    0.222    0.222
## 20 EIB1 ~~ EIB3 40.862  0.066   0.066    0.241    0.241
```

第二次修正

```
valid_EIB3 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
EIB5    ~~  EIB7
EIB1    ~~  EIB2
"
```

```
model_valid_EIB3 <- cfa(valid_EIB3,
                        data = df1)
fitMeasures(model_valid_EIB3,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr     gfi
## 296.835  18.000   0.000   0.948   0.124   0.920   0.043   0.924
```

```
modindices(model_valid_EIB3, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 45 EIB7 ~~ EIB8 51.484  0.089   0.089    0.256    0.256
## 20 EIB1 ~~ EIB3 44.023  0.068   0.068    0.226    0.226
## 43 EIB6 ~~ EIB7 34.988  0.080   0.080    0.204    0.204
## 23 EIB1 ~~ EIB6 33.538 -0.066  -0.066   -0.201   -0.201
## 39 EIB4 ~~ EIB7 32.804 -0.069  -0.069   -0.208   -0.208
```

```
valid_EIB4 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
EIB5    ~~  EIB7
EIB1    ~~  EIB2
EIB7    ~~  EIB8
"
model_valid_EIB4 <- cfa(valid_EIB4,
                        data = df1)
fitMeasures(model_valid_EIB4,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr     gfi
## 245.635  17.000   0.000   0.958   0.115   0.930   0.040   0.939
```

```
modindices(model_valid_EIB4, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
```

```
## 44 EIB6 ~~ EIB7 42.115  0.088   0.088    0.210    0.210
## 21 EIB1 ~~ EIB3 38.415  0.063   0.063    0.218    0.218
## 24 EIB1 ~~ EIB6 33.599 -0.067  -0.067   -0.204   -0.204
## 35 EIB3 ~~ EIB6 27.070 -0.065  -0.065   -0.196   -0.196
## 42 EIB5 ~~ EIB6 25.787  0.073   0.073    0.173    0.173
```

第四次修正

```
valid_EIB5 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
EIB5    ~~  EIB7
EIB1    ~~  EIB2
EIB7    ~~  EIB8
EIB6    ~~  EIB7
"
model_valid_EIB5 <- cfa(valid_EIB5,
                        data = df1)
fitMeasures(model_valid_EIB5,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli    srmr     gfi
## 205.441  16.000   0.000   0.965   0.108   0.939   0.036   0.948
```

```
modindices(model_valid_EIB5, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 43 EIB5 ~~ EIB6 67.400  0.129   0.129    0.302    0.302
## 22 EIB1 ~~ EIB3 36.657  0.062   0.062    0.215    0.215
## 25 EIB1 ~~ EIB6 35.144 -0.067  -0.067   -0.202   -0.202
## 45 EIB6 ~~ EIB8 29.025  0.072   0.072    0.208    0.208
## 35 EIB3 ~~ EIB5 26.099 -0.066  -0.066   -0.178   -0.178
```

第五次修正

```
valid_EIB6 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
```

```
EIB5    ~~  EIB7
EIB1    ~~  EIB2
EIB7    ~~  EIB8
EIB6    ~~  EIB7
EIB5    ~~  EIB6
"
model_valid_EIB6 <- cfa(valid_EIB6,
                        data = df1)
fitMeasures(model_valid_EIB6,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##   chisq      df  pvalue     cfi   rmsea     tli     srmr     gfi
## 138.602  15.000   0.000   0.977   0.090   0.957   0.029   0.966
```

```
modindices(model_valid_EIB6, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 45 EIB6 ~~ EIB8 40.958  0.083   0.083    0.227    0.227
## 23 EIB1 ~~ EIB3 27.098  0.053   0.053    0.195    0.195
## 26 EIB1 ~~ EIB6 26.308 -0.056  -0.056   -0.164   -0.164
## 39 EIB3 ~~ EIB8 10.912 -0.037  -0.037   -0.125   -0.125
## 29 EIB2 ~~ EIB3  9.620  0.032   0.032    0.118    0.118
```

第六次修正

```
valid_EIB7 <- "v_EIB =~ EIB1 + EIB2 +
EIB3 + EIB4 + EIB5 +EIB6 + EIB7 + EIB8
EIB5    ~~  EIB7
EIB1    ~~  EIB2
EIB7    ~~  EIB8
EIB6    ~~  EIB7
EIB5    ~~  EIB6
EIB6    ~~  EIB8
"
```

```
model_valid_EIB7 <- cfa(valid_EIB7,
                        data = df1)
fitMeasures(model_valid_EIB7,
            c("chisq", "df", "pvalue","cfi",
              "rmsea", "tli", "srmr", "gfi"))
```

```
##  chisq     df pvalue    cfi  rmsea    tli   srmr    gfi
## 99.127 14.000  0.000  0.984  0.077  0.969  0.024  0.975
```

```
modindices(model_valid_EIB7, sort = T, maximum.number = 5)
```

```
##     lhs op  rhs     mi    epc sepc.lv sepc.all sepc.nox
## 24 EIB1 ~~ EIB3 20.209  0.046   0.046    0.176    0.176
## 45 EIB5 ~~ EIB8 20.098  0.070   0.070    0.168    0.168
## 27 EIB1 ~~ EIB6 15.599 -0.042  -0.042   -0.123   -0.123
## 42 EIB4 ~~ EIB6 14.991  0.045   0.045    0.138    0.138
## 37 EIB3 ~~ EIB5 14.790 -0.047  -0.047   -0.128   -0.128
```

```
df1 %>% dplyr::select(EIB1:EIB8) %>%
  KMO()
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = .)
## Overall MSA =  0.93
## MSA for each item =
## EIB1 EIB2 EIB3 EIB4 EIB5 EIB6 EIB7 EIB8
## 0.92 0.93 0.93 0.94 0.93 0.93 0.91 0.94
```

```
df1 %>% dplyr::select(EIB1:EIB8) %>%
  cortest.bartlett()
```

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

```
## $chisq
## [1] 5410.361
##
## $p.value
```

```
## [1] 0
##
## $df
## [1] 28
```

```
df1 %>% dplyr::select(EIB1:EIB8) %>%
  principal(nfactors=1, score=TRUE)
```

```
## Principal Components Analysis
## Call: principal(r = ., nfactors = 1, scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##       PC1   h2   u2 com
## EIB1 0.81 0.66 0.34   1
## EIB2 0.83 0.69 0.31   1
## EIB3 0.78 0.61 0.39   1
## EIB4 0.84 0.70 0.30   1
## EIB5 0.78 0.62 0.38   1
## EIB6 0.80 0.64 0.36   1
## EIB7 0.80 0.64 0.36   1
## EIB8 0.83 0.69 0.31   1
##
##                 PC1
## SS loadings    5.25
## Proportion Var 0.66
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.08
##  with the empirical chi square  320.2  with prob <  5.9e-56
##
## Fit based upon off diagonal values = 0.98
```

```
semTools::reliability(model_valid_EIB)
```

```
##            v_EIB
## alpha  0.9242452
## omega  0.9245129
## omega2 0.9245129
## omega3 0.9238404
## avevar 0.6056565
```

4 描述性统计和相关性分析

4.1 数据各个维度加总

```
df2 <- df1 %>%
  mutate(
    SNS_t = (SNS1 + SNS2 + SNS3 +
               SNS4 + SNS5 + SNS6) / 6,
    SES_t = SES1 + SES2 + SES3_r +
      SES4 + SES5_r + SES6 + SES7 +
      SES8 + SES9_r + SES10_r,
    CSES_t = CSES1 + CSES2 +
      CSES3 + CSES4,
    EIB_t = EIB1 + EIB2 + EIB3 +
      EIB4 + EIB5 +
      EIB6 + EIB7 + EIB8
  ) %>%
  dplyr::select(
    id, IP, sex, grade,
    residence, school,
    SNS_t, SES_t,
    CSES_t, EIB_t
  ) %>% mutate(sex = if_else(sex == 1, "male", "female")) %>%
```

```
  mutate(residence = if_else(residence == 1, "city", "rural")) %>%
  mutate(school = if_else(school == 1,  "public", "private")) %>%
  mutate(grade = case_when(
    grade == 1 ~ "freshman",
    grade == 2 ~ "sophomore",
    grade == 3 ~ "junior",
    grade == 4 ~ "senior"
  )) %>% mutate(across(IP:school, as.factor))

df2 <- df2 %>%
  mutate(inter_raw = (SES_t - mean(SES_t)) * SNS_t,
                      center_SES = scale(SES_t, center = TRUE,
                                         scale = FALSE)) %>% mutate(
    IP = str_extract(IP, "[\u4e00-\u9fa5]+")
  )
####3 虚拟编码
df2 <- cbind(df2, dummy.code(df2$sex))
df2 <- cbind(df2, dummy.code(df2$grade))
df2 <- cbind(df2, dummy.code(df2$residence))
df2 <- cbind(df2, dummy.code(df2$school))
```

转换数据类型

```
# df2$sex <- factor(df2$sex)
# df2$grade <- factor(df2$grade)
# df2$school <- factor(df2$school)
df2$inter_raw <- as.numeric(df2$inter_raw)
df2$center_SES <- as.numeric(df2$center_SES)
df2$SNS_t <- as.numeric(df2$SNS_t)
str(df2)
```

```
## 'data.frame':    1014 obs. of  22 variables:
##  $ id         : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ IP         : chr  "云南" "河北" "云南" "云南" ...
```

```
##  $ sex       : Factor w/ 2 levels "female","male": 2 2 1 1 1 2 1 1 1 2 ...
##  $ grade     : Factor w/ 4 levels "freshman","junior",..: 4 3 3 4 4 4 1 4 4 4 ...
##  $ residence : Factor w/ 2 levels "city","rural": 1 1 2 1 1 2 2 1 2 1 ...
##  $ school    : Factor w/ 2 levels "private","public": 1 1 2 1 1 1 2 1 2 1 ...
##  $ SNS_t     : num  0.8186 0.1317 0.3727 0.0709 -2.0149 ...
##  $ SES_t     : num  31 34 28 28 31 35 31 25 33 37 ...
##  $ CSES_t    : num  25 22 17 22 22 28 15 16 18 28 ...
##  $ EIB_t     : num  38 29 27 30 30 35 25 16 22 38 ...
##  $ inter_raw : num  1.9996 0.7168 -0.2076 -0.0395 -4.922 ...
##  $ center_SES: num  2.443 5.443 -0.557 -0.557 2.443 ...
##  $ female    : num  0 0 1 1 1 0 1 1 1 0 ...
##  $ male      : num  1 1 0 0 0 1 0 0 0 1 ...
##  $ freshman  : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ sophomore : num  1 0 0 1 1 1 0 1 1 1 ...
##  $ junior    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ senior    : num  0 1 1 0 0 0 0 0 0 0 ...
##  $ rural     : num  0 0 1 0 0 1 1 0 1 0 ...
##  $ city      : num  1 1 0 1 1 0 0 1 0 1 ...
##  $ private   : num  1 1 0 1 1 1 0 1 0 1 ...
##  $ public    : num  0 0 1 0 0 0 1 0 1 0 ...
```

写出数据（如果需要的话）

```
# write_xlsx(df2, " 清洗完的数据.xlsx")
# write_csv(df2, " 清洗完的数据.csv")
# write_csv(df2, "abc.csv")
```

4.2 描述性统计

```
df_report <- df2 %>%
  dplyr::select(SNS_t, SES_t, CSES_t, EIB_t, IP, sex, grade, residence, school)

report::report(df_report)
```

```
## The data contains 1014 observations of the following 9 variables:
##   - SNS_t: n = 1014, Mean = 8.55e-17, SD = 0.79, Median = 0.07, MAD = 0.86, range: [
##   - SES_t: n = 1014, Mean = 28.56, SD = 4.46, Median = 28.00, MAD = 2.97, range: [11
##   - CSES_t: n = 1014, Mean = 19.37, SD = 4.68, Median = 20.00, MAD = 4.45, range: [4
##   - EIB_t: n = 1014, Mean = 28.49, SD = 5.90, Median = 28.00, MAD = 5.93, range: [8,
##   - IP: 29 entries, such as 云南 (59.27%); 广东 (11.05%); 浙江 (9.27%) and 26 others
##   - sex: 2 levels, namely female (n = 536, 52.86%) and male (n = 478, 47.14%)
##   - grade: 4 levels, namely freshman (n = 370, 36.49%), junior (n = 150, 14.79%), se
##   - residence: 2 levels, namely city (n = 486, 47.93%) and rural (n = 528, 52.07%)
##   - school: 2 levels, namely private (n = 599, 59.07%) and public (n = 415, 40.93%)
```

```
df_report %>%
  group_by(sex) %>%
  summarise(mean = mean(EIB_t))
```

```
## # A tibble: 2 x 2
##   sex     mean
##   <fct>  <dbl>
## 1 female  27.8
## 2 male    29.2
```

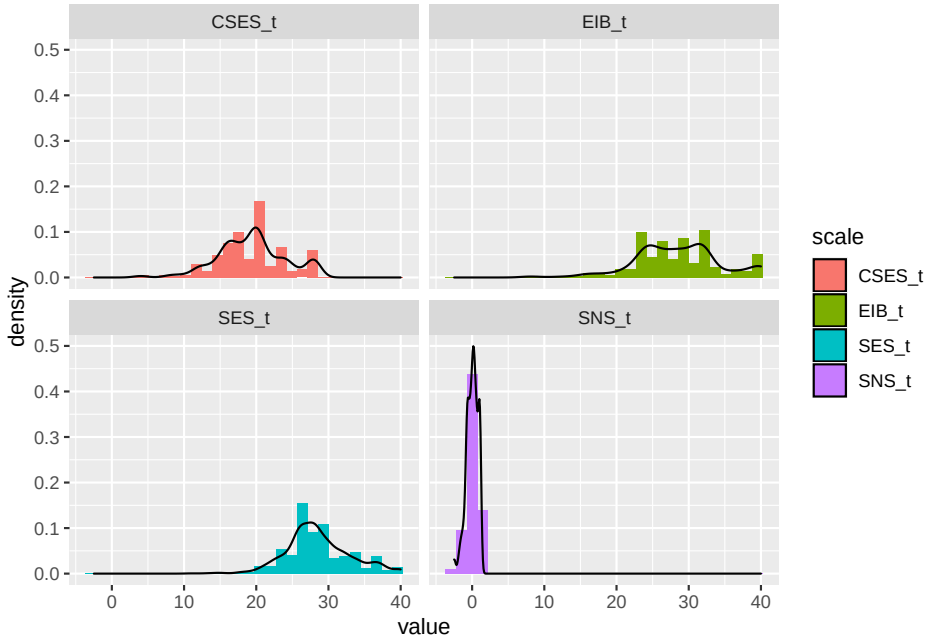
```
df_report %>%
  group_by(grade) %>%
  summarise(mean = mean(EIB_t))
```

```
## # A tibble: 4 x 2
##   grade      mean
##   <fct>     <dbl>
## 1 freshman   28.0
## 2 junior     29.4
## 3 senior     29.7
## 4 sophomore  28.2
```

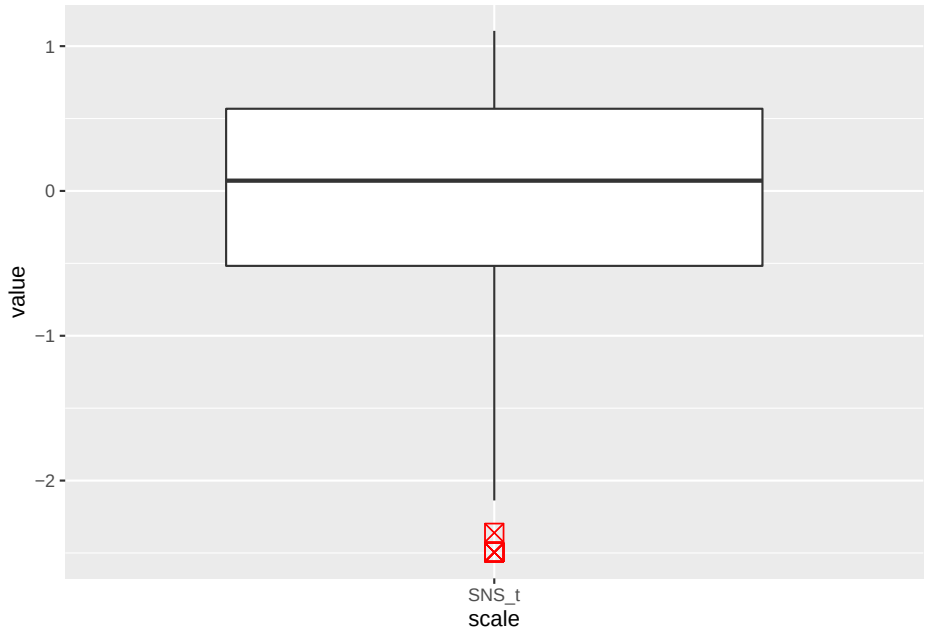
4.3 可视化

```
df2 %>%
  select(id, SNS_t, SES_t, CSES_t, EIB_t) %>%
  pivot_longer(cols = SNS_t:EIB_t, names_to = "scale",
               values_to = "value") %>%
   ggplot(aes(x = value, y = stat(density))) +
  geom_histogram(aes(fill = scale)) +
  geom_density() +
  facet_wrap(~scale)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
df2 %>%
  select(id, SNS_t, SES_t, CSES_t, EIB_t) %>%
  pivot_longer(cols = SNS_t:EIB_t, names_to = "scale",
               values_to = "value") %>% filter(scale == "SNS_t") %>%
   ggplot(aes(x = scale, y = value)) +
  geom_boxplot(outlier.colour="red", outlier.shape=7,outlier.size=4)
```



4.4 相关分析

```
cor(df_report[1:4])
```

```
##            SNS_t     SES_t    CSES_t     EIB_t
## SNS_t  1.0000000 0.1561484 0.3587445 0.3267670
## SES_t  0.1561484 1.0000000 0.4936694 0.4866594
## CSES_t 0.3587445 0.4936694 1.0000000 0.7925201
## EIB_t  0.3267670 0.4866594 0.7925201 1.0000000
```

```
cor_p <- rcorr(as.matrix(df_report[1:4]))
cor_p
```

```
##        SNS_t SES_t CSES_t EIB_t
## SNS_t   1.00  0.16   0.36  0.33
## SES_t   0.16  1.00   0.49  0.49
## CSES_t  0.36  0.49   1.00  0.79
## EIB_t   0.33  0.49   0.79  1.00
```

```
##
## n= 1014
##
##
## P
##        SNS_t SES_t CSES_t EIB_t
## SNS_t         0     0      0
## SES_t   0           0      0
## CSES_t  0     0            0
## EIB_t   0     0     0
```

4.4.1 筛选控制变量

```
t.test(EIB_t ~ sex, df2)# 显著
```

4.4.1.1 性别显著

```
##
##  Welch Two Sample t-test
##
## data:  EIB_t by sex
## t = -3.7686, df = 994.38, p-value = 0.0001738
## alternative hypothesis: true difference in means between group female and group male
## 95 percent confidence interval:
##  -2.1164584 -0.6670613
## sample estimates:
## mean in group female   mean in group male
##             27.83209             29.22385
```

```
cohens_d(EIB_t ~ sex, data =df2)
```

```
## Cohen's d |         95% CI
## -------------------------
## -0.24     | [-0.36, -0.12]
```

```
##
## - Estimated using pooled SD.
```

```
summary(aovsex <- aov(EIB_t ~ sex, data=df2))
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## sex            1    489   489.4   14.23 0.000171 ***
## Residuals   1012  34800    34.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
eta_squared(aovsex)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to eta squar
## Returning eta squared.
```

```
## # Effect Size for ANOVA
##
## Parameter | Eta2 |       95% CI
## ------------------------------
## sex       | 0.01 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

```
summary(g_m <- aov(EIB_t ~ grade, df2))
```

4.4.1.2 年级显著

```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## grade          3    427  142.20    4.12 0.00645 **
## Residuals   1010  34863   34.52
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
eta_squared(g_m)
```

```
## For one-way between subjects designs, partial eta squared is equivalent to eta squar
## Returning eta squared.
```

```
## # Effect Size for ANOVA
##
## Parameter | Eta2 |       95% CI
## ------------------------------
## grade     | 0.01 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

```
t.test(EIB_t ~ residence, df2)
```

4.4.1.3 户口不显著

```
##
##  Welch Two Sample t-test
##
## data:  EIB_t by residence
## t = 1.0285, df = 995.43, p-value = 0.304
## alternative hypothesis: true difference in means between group city and group rural
## 95 percent confidence interval:
##  -0.3471313  1.1117684
## sample estimates:
##  mean in group city mean in group rural
##            28.68724            28.30492
```

```
cohens_d(EIB_t ~ residence, data =df2)
```

```
## Cohen's d |        95% CI
## -------------------------
## 0.06      | [-0.06, 0.19]
```

```
##
## - Estimated using pooled SD.
```

```
t.test(EIB_t ~ school, df2)
```

4.4.1.4 学校性质显著

```
##
##  Welch Two Sample t-test
##
## data:  EIB_t by school
## t = -2.7933, df = 863.81, p-value = 0.005333
## alternative hypothesis: true difference in means between group private and group pub
## 95 percent confidence interval:
##  -1.8016847 -0.3146376
## sample estimates:
## mean in group private  mean in group public
##              28.05509              29.11325
```

```
cohens_d(EIB_t ~ school, data =df2)
```

```
## Cohen's d |         95% CI
## -------------------------
## -0.18     | [-0.31, -0.05]
##
## - Estimated using pooled SD.
```

```
aov(EIB_t ~ school, data=df2) %>% summary()
```

```
##               Df Sum Sq Mean Sq F value  Pr(>F)
## school         1    274   274.5   7.934 0.00495 **
## Residuals   1012  35015    34.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
aov(EIB_t ~ school, data=df2) %>% eta_squared()
```

```
## For one-way between subjects designs, partial eta squared is equivalent to eta squar
## Returning eta squared.
```

```
## # Effect Size for ANOVA
##
## Parameter |     Eta2 |       95% CI
## -----------------------------------
## school    | 7.78e-03 | [0.00, 1.00]
##
## - One-sided CIs: upper bound fixed at (1).
```

5 间接效应

5.1 有调节的中介

5.1.1 简单中介检验

```
simple_med <- '
EIB_t ~ b1*CSES_t
EIB_t ~ cdash*SNS_t
CSES_t ~ a1*SNS_t
### 控制变量
EIB_t ~ male
EIB_t ~ public
EIB_t ~ freshman
EIB_t ~ sophomore
EIB_t ~ junior

CSES_t ~ male
CSES_t ~ public
```

```
CSES_t ~ freshman
CSES_t ~ sophomore
CSES_t ~ junior
#ind and total
ind := a1*b1
direct := b1

total := a1*b1 + cdash
'

fit_simple_med <- sem(simple_med, data = df2,
                      se = "bootstrap",bootstrap = 5000)
parameterEstimates(fit_simple_med , standardized = TRUE,
                   rsquare = T, output = "text", header = TRUE)
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws             5000
##   Number of successful bootstrap draws            5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   EIB_t ~
##     CSES_t    (b1)    0.967    0.029   33.047    0.000    0.908    1.024
##     SNS_t   (cdsh)    0.376    0.170    2.214    0.027    0.045    0.706
##   CSES_t ~
##     SNS_t     (a1)    2.116    0.214    9.899    0.000    1.691    2.536
##   EIB_t ~
##     male              0.492    0.227    2.164    0.030    0.045    0.928
##     public            0.262    0.257    1.018    0.309   -0.226    0.780
##     freshmn          -0.126    0.386   -0.327    0.744   -0.873    0.654
```

```
##     sophomr          -0.400    0.372   -1.075    0.282   -1.127    0.350
##     junior           -0.084    0.449   -0.188    0.851   -0.953    0.796
##   CSES_t ~
##     male              1.390    0.277    5.017    0.000    0.861    1.948
##     public            0.073    0.296    0.246    0.806   -0.529    0.643
##     freshmn          -0.784    0.509   -1.541    0.123   -1.791    0.181
##     sophomr          -0.760    0.499   -1.522    0.128   -1.774    0.235
##     junior           -0.028    0.554   -0.051    0.959   -1.143    1.052
##    Std.lv  Std.all
##
##     0.967    0.767
##     0.376    0.050
##
##     2.116    0.358
##
##     0.492    0.042
##     0.262    0.022
##    -0.126   -0.010
##    -0.400   -0.033
##    -0.084   -0.005
##
##     1.390    0.148
##     0.073    0.008
##    -0.784   -0.081
##    -0.760   -0.078
##    -0.028   -0.002
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .EIB_t           12.784    0.757   16.891    0.000   11.284   14.270
##    .CSES_t          18.550    0.823   22.539    0.000   16.845   20.061
##    Std.lv  Std.all
##    12.784    0.367
```

```
##    18.550    0.847
##
## R-Square:
##                    Estimate
##     EIB_t             0.633
##     CSES_t            0.153
##
## Defined Parameters:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##     ind               2.046    0.228    8.968    0.000    1.608    2.508
##     direct            0.967    0.029   33.043    0.000    0.908    1.024
##     total             2.421    0.285    8.499    0.000    1.861    2.983
##    Std.lv  Std.all
##     2.046    0.275
##     0.967    0.767
##     2.421    0.325
```

总效应

```
direct_mod <- lm(scale(df2$EIB_t, center = T,
                       scale = T) ~
                   scale(df2$SNS_t, center = T, scale = T) +
                   male + freshman + sophomore + junior + public,
                 df2)

med_mod <- lm(scale(df2$CSES_t, center = T,
                    scale = T) ~
                scale(df2$SNS_t, center = T, scale = T)+
                male + freshman + sophomore + junior + public,
              df2)

med_mod2 <- lm(scale(df2$EIB_t, center = T,
                     scale = T) ~
                 scale(df2$CSES_t, center = T, scale = T)+
```

```
                male + freshman + sophomore + junior + public,
              df2)

out_mod <- lm(scale(df2$EIB_t, center = T,
                    scale = T) ~
                scale(df2$SNS_t, center = T, scale = T) +
                scale(df2$CSES_t, center = T, scale = T)+
                male + freshman + sophomore + junior + public,
              df2)

summary(direct_mod)
```

```
##
## Call:
## lm(formula = scale(df2$EIB_t, center = T, scale = T) ~ scale(df2$SNS_t,
##     center = T, scale = T) + male + freshman + sophomore + junior +
##     public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.10359 -0.58330  0.01053  0.60108  2.24681
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -0.04210    0.09851  -0.427   0.6692
## scale(df2$SNS_t, center = T, scale = T)   0.32546    0.03008  10.821  < 2e-16
## male                                      0.31102    0.05942   5.234 2.02e-07
## freshman                                 -0.14972    0.10293  -1.454   0.1461
## sophomore                                -0.19215    0.09870  -1.947   0.0518
## junior                                   -0.01894    0.11338  -0.167   0.8674
## public                                    0.05627    0.06504   0.865   0.3872
##
## (Intercept)
```

```
## scale(df2$SNS_t, center = T, scale = T) ***
## male                                    ***
## freshman
## sophomore                               .
## junior
## public
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9331 on 1007 degrees of freedom
## Multiple R-squared:  0.1345, Adjusted R-squared:  0.1293
## F-statistic: 26.07 on 6 and 1007 DF,  p-value: < 2.2e-16
```

```
summary(med_mod)
```

```
##
## Call:
## lm(formula = scale(df2$CSES_t, center = T, scale = T) ~ scale(df2$SNS_t,
##     center = T, scale = T) + male + freshman + sophomore + junior +
##     public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.08767 -0.56920 -0.01531  0.60558  2.43906
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -0.025154   0.097429  -0.258   0.7963
## scale(df2$SNS_t, center = T, scale = T)  0.358475   0.029747  12.051  < 2e-16
## male                                     0.296848   0.058774   5.051 5.22e-07
## freshman                                -0.167337   0.101808  -1.644   0.1006
## sophomore                               -0.162217   0.097615  -1.662   0.0969
## junior                                  -0.006031   0.112142  -0.054   0.9571
## public                                   0.015548   0.064329   0.242   0.8091
```

```
##
## (Intercept)
## scale(df2$SNS_t, center = T, scale = T) ***
## male                                    ***
## freshman
## sophomore                               .
## junior
## public
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9229 on 1007 degrees of freedom
## Multiple R-squared:  0.1533, Adjusted R-squared:  0.1483
## F-statistic: 30.39 on 6 and 1007 DF,  p-value: < 2.2e-16
```

```
summary(med_mod2)
```

```
##
## Call:
## lm(formula = scale(df2$EIB_t, center = T, scale = T) ~ scale(df2$CSES_t,
##     center = T, scale = T) + male + freshman + sophomore + junior +
##     public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51937 -0.32943  0.03257  0.45049  2.70631
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -0.01592    0.06430  -0.248   0.8045
## scale(df2$CSES_t, center = T, scale = T)  0.78483    0.01946  40.330   <2e-16
## male                                     0.07349    0.03910   1.880   0.0604
## freshman                                -0.03387    0.06714  -0.504   0.6141
## sophomore                               -0.07265    0.06454  -1.126   0.2605
```

```
## junior                                      -0.01782    0.07406  -0.241   0.8099
## public                                       0.05567    0.04223   1.318   0.1877
##
## (Intercept)
## scale(df2$CSES_t, center = T, scale = T) ***
## male                                        .
## freshman
## sophomore
## junior
## public
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6096 on 1007 degrees of freedom
## Multiple R-squared:  0.6306, Adjusted R-squared:  0.6284
## F-statistic: 286.5 on 6 and 1007 DF,  p-value: < 2.2e-16
```

```
summary(out_mod)
```

```
##
## Call:
## lm(formula = scale(df2$EIB_t, center = T, scale = T) ~ scale(df2$SNS_t,
##     center = T, scale = T) + scale(df2$CSES_t, center = T, scale = T) +
##     male + freshman + sophomore + junior + public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52305 -0.33599  0.05121  0.42914  2.64029
##
## Coefficients:
##                                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                -0.02280    0.06421  -0.355   0.7225
## scale(df2$SNS_t, center = T, scale = T)     0.05048    0.02097   2.407   0.0163
## scale(df2$CSES_t, center = T, scale = T)    0.76708    0.02077  36.938   <2e-16
```

```
## male                                          0.08331    0.03922   2.124   0.0339
## freshman                                     -0.02136    0.06718  -0.318   0.7506
## sophomore                                    -0.06772    0.06442  -1.051   0.2934
## junior                                       -0.01431    0.07390  -0.194   0.8464
## public                                        0.04434    0.04239   1.046   0.2958
##
## (Intercept)
## scale(df2$SNS_t, center = T, scale = T)  *
## scale(df2$CSES_t, center = T, scale = T) ***
## male                                     *
## freshman
## sophomore
## junior
## public
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6082 on 1006 degrees of freedom
## Multiple R-squared:  0.6327, Adjusted R-squared:  0.6301
## F-statistic: 247.5 on 7 and 1006 DF,  p-value: < 2.2e-16
```

```
simple <- '
EIB_t ~ SNS_t + male + freshman + sophomore + junior + public
'
fit_simple <- sem(simple, data = df2, se = "bootstrap",
                  bootstrap = 5000)
summary(fit_simple)
```

```
## lavaan 0.6-10 ended normally after 1 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         7
##
```

```
##   Number of observations                          1014
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws              5000
##   Number of successful bootstrap draws             5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EIB_t ~
##     SNS_t             2.421    0.290    8.355    0.000
##     male              1.836    0.356    5.163    0.000
##     freshman         -0.884    0.642   -1.377    0.168
##     sophomore        -1.134    0.623   -1.820    0.069
##     junior           -0.112    0.714   -0.156    0.876
##     public            0.332    0.391    0.849    0.396
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .EIB_t            30.123    1.369   21.997    0.000
```

5.1.2 有调节的中介

```
moderated_model <- '
EIB_t ~ b1*CSES_t
EIB_t ~ cdash*SNS_t
EIB_t ~ male
```

```
EIB_t ~ public
EIB_t ~ freshman
EIB_t ~ sophomore
EIB_t ~ junior
CSES_t ~ a1*SNS_t
CSES_t ~ a2*center_SES
CSES_t ~ a3*inter_raw
CSES_t ~ male
CSES_t ~ public
CSES_t ~ freshman
CSES_t ~ sophomore
CSES_t ~ junior
# 间接效应,Conditional Indirect Effect
ind_low := a1*b1 - a3*b1*4.455127 # 低分组简单效应
ind_med := a1*b1 - a3*b1*0
ind_high := a1*b1 + a3*b1*4.455127
# 之间的差异
dif1 := ind_med - ind_low
dif2 := ind_high -ind_low
dif3 := ind_high - ind_med
# 直接效应和有调节的中介效应
imm := a3*b1
direct := cdash # 直接效应
# 总效应
total_low := cdash +ind_low
total_med := cdash +ind_med
total_high := cdash +ind_high
'
```

5.1.3 模型结果

```
fit_mod_model <- sem(moderated_model, data = df2,
                     se = "bootstrap",bootstrap = 5000)
parameterEstimates(fit_mod_model, standardized = TRUE,
                   rsquare = T, output = "text", header = TRUE)
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws              5000
##   Number of successful bootstrap draws             5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   EIB_t ~
##     CSES_t    (b1)    0.967    0.030   32.725    0.000    0.907    1.023
##     SNS_t   (cdsh)    0.376    0.168    2.237    0.025    0.044    0.707
##     male               0.492    0.224    2.196    0.028    0.053    0.920
##     public             0.262    0.250    1.045    0.296   -0.239    0.749
##     freshmn           -0.126    0.382   -0.330    0.742   -0.898    0.608
##     sophomr           -0.400    0.371   -1.078    0.281   -1.143    0.311
##     junior            -0.084    0.445   -0.190    0.849   -0.955    0.784
##   CSES_t ~
##     SNS_t     (a1)    1.704    0.179    9.538    0.000    1.354    2.044
##     cnt_SES   (a2)    0.468    0.030   15.465    0.000    0.409    0.525
##     intr_rw   (a3)   -0.073    0.035   -2.059    0.040   -0.143   -0.006
##     male               1.107    0.243    4.550    0.000    0.606    1.584
##     public            -0.022    0.265   -0.083    0.934   -0.548    0.495
##     freshmn           -0.602    0.436   -1.379    0.168   -1.465    0.246
##     sophomr           -0.182    0.424   -0.431    0.667   -1.005    0.633
##     junior            -0.108    0.474   -0.229    0.819   -1.070    0.833
##    Std.lv  Std.all
##
```

```
##     0.967    0.767
##     0.376    0.050
##     0.492    0.042
##     0.262    0.022
##    -0.126   -0.010
##    -0.400   -0.033
##    -0.084   -0.005
##
##     1.704    0.289
##     0.468    0.445
##    -0.073   -0.062
##     1.107    0.118
##    -0.022   -0.002
##    -0.602   -0.062
##    -0.182   -0.019
##    -0.108   -0.008
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .EIB_t           12.784    0.760   16.812    0.000   11.232   14.175
##    .CSES_t          14.362    0.668   21.488    0.000   12.909   15.582
##    Std.lv  Std.all
##    12.784    0.367
##    14.362    0.656
##
## R-Square:
##                   Estimate
##     EIB_t            0.633
##     CSES_t           0.344
##
## Defined Parameters:
##                   Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##     ind_low          1.960    0.257    7.624    0.000    1.458    2.472
```

```
##     ind_med           1.647    0.187    8.804    0.000    1.280    2.011
##     ind_high          1.334    0.225    5.927    0.000    0.896    1.772
##     dif1             -0.313    0.153   -2.045    0.041   -0.626   -0.025
##     dif2             -0.625    0.306   -2.045    0.041   -1.253   -0.050
##     dif3             -0.313    0.153   -2.045    0.041   -0.626   -0.025
##     imm              -0.070    0.034   -2.045    0.041   -0.141   -0.006
##     direct            0.376    0.168    2.237    0.025    0.044    0.707
##     total_low         2.335    0.309    7.560    0.000    1.739    2.939
##     total_med         2.023    0.241    8.397    0.000    1.540    2.497
##     total_high        1.710    0.260    6.587    0.000    1.196    2.214
##    Std.lv  Std.all
##     1.960    0.432
##     1.647    0.221
##     1.334    0.011
##    -0.313   -0.211
##    -0.625   -0.422
##    -0.313   -0.211
##    -0.070   -0.047
##     0.376    0.050
##     2.335    0.483
##     2.023    0.272
##     1.710    0.061
```

5.2 模型比较

```
fitmeasures(fit_mod_model, c("aic", "ecvi", "bic"))
```

```
##       aic      ecvi       bic
## 11074.963     0.067 11158.631
```

5.2.1 简单斜率分析

```
simple_slope_m <- lm(CSES_t ~ center_SES*SNS_t + male + freshman +
                       sophomore + junior + public , data = df2)

summary(simple_slope_m)
```

```
##
## Call:
## lm(formula = CSES_t ~ center_SES * SNS_t + male + freshman +
##     sophomore + junior + public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.4763  -2.4183  -0.0279   2.3518  11.9704
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      19.19801    0.40309  47.627  < 2e-16 ***
## center_SES        0.46776    0.02747  17.028  < 2e-16 ***
## SNS_t             1.70365    0.15655  10.882  < 2e-16 ***
## male              1.10718    0.24309   4.555 5.89e-06 ***
## freshman         -0.60152    0.42044  -1.431   0.1528
## sophomore        -0.18244    0.40429  -0.451   0.6519
## junior           -0.10847    0.46262  -0.234   0.8147
## public           -0.02192    0.26540  -0.083   0.9342
## center_SES:SNS_t -0.07258    0.03017  -2.406   0.0163 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.807 on 1005 degrees of freedom
## Multiple R-squared:  0.3445, Adjusted R-squared:  0.3393
## F-statistic: 66.02 on 8 and 1005 DF,  p-value: < 2.2e-16
```

```
model_2nd <- lm(scale(EIB_t) ~ scale(SNS_t) + scale(CSES_t) + scale(male) + scale(fresh
                  scale(junior) + scale(public), data = df2)
summary(model_2nd)
```

```
##
## Call:
## lm(formula = scale(EIB_t) ~ scale(SNS_t) + scale(CSES_t) + scale(male) +
##     scale(freshman) + scale(sophomore) + scale(junior) + scale(public),
##     data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52305 -0.33599  0.05121  0.42914  2.64029
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.372e-16  1.910e-02   0.000   1.0000
## scale(SNS_t)      5.048e-02  2.097e-02   2.407   0.0163 *
## scale(CSES_t)     7.671e-01  2.077e-02  36.938   <2e-16 ***
## scale(male)       4.161e-02  1.959e-02   2.124   0.0339 *
## scale(freshman)  -1.029e-02  3.236e-02  -0.318   0.7506
## scale(sophomore) -3.261e-02  3.103e-02  -1.051   0.2934
## scale(junior)    -5.085e-03  2.625e-02  -0.194   0.8464
## scale(public)     2.181e-02  2.086e-02   1.046   0.2958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6082 on 1006 degrees of freedom
## Multiple R-squared:  0.6327, Adjusted R-squared:  0.6301
## F-statistic: 247.5 on 7 and 1006 DF,  p-value: < 2.2e-16
```

```
sd(df2$center_SES)
```

```
## [1] 4.455127
```

```
library(showtext)
```

```
## 载入需要的程辑包: sysfonts
```

```
## 载入需要的程辑包: showtextdb
```

```
showtext_auto()
library(effects)
```

```
## 载入需要的程辑包: carData
```

```
## Use the command
##     lattice::trellis.par.set(effectsTheme())
##   to customize lattice options for effects plots.
## See ?effectsTheme for details.
```

```
Inter.1a<-effect(c("center_SES:SNS_t"),
                 simple_slope_m,
                 xlevels=list(center_SES=c(
                   -4.455127,0, 4.455127)))
font_families()
```

```
## [1] "sans"         "serif"        "mono"         "wqy-microhei"
```

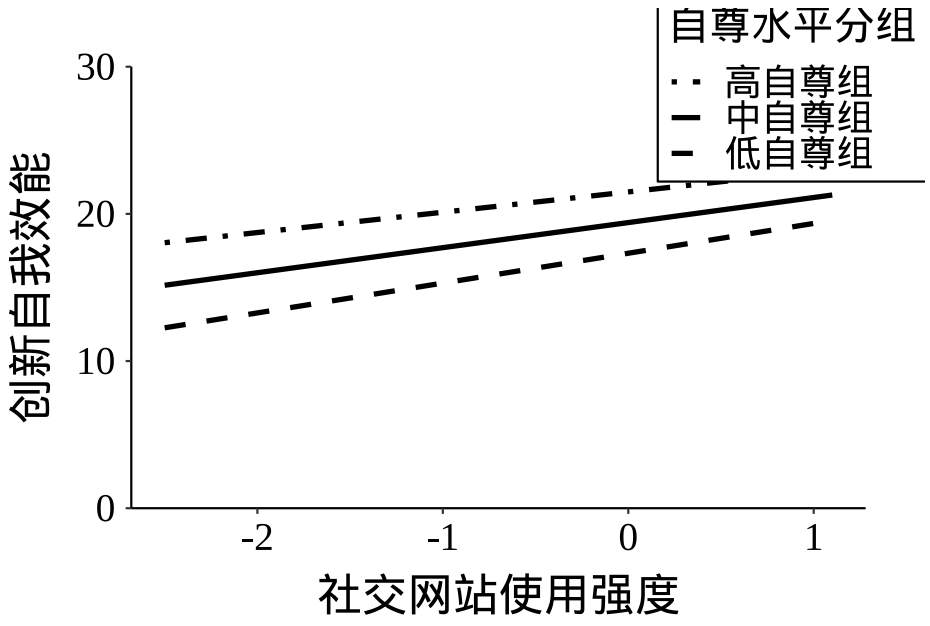
```
df3 <- as.data.frame(Inter.1a)
new <- "serif"
df3$center_SES <- ifelse(df3$center_SES == -4.455127,
                         " 低自尊组",
                         ifelse(df3$center_SES ==4.455127,
                                " 高自尊组",
                                " 中自尊组"))

df3$center_SES <- factor(df3$center_SES,
                         levels = c(" 高自尊组", " 中自尊组",
                                    " 低自尊组"))

df3 <- df3 %>% rename(" 自尊水平分组" = center_SES)
```

```
ggplot(data = df3, aes(x = SNS_t, y = fit,
                       group = 自尊水平分组,
                       shape = 自尊水平分组,
                       linetype = 自尊水平分组)) +
  geom_line(size = 1.2) +
  scale_linetype_manual(
    values = c('dotdash', 'solid', 'dashed'))+
  coord_cartesian(ylim = c(0, 30)) +
  scale_y_continuous(expand = expansion(0))+
  theme_classic() +
  labs(
   x = " 社交网站使用强度",
   y = " 创新自我效能",
   fill = NULL,
   title = NULL
 )+
  theme(
  plot.margin = unit(c(1, 1, 1, 1), "cm"),
    panel.background = element_blank(),
    plot.title = element_text(size = 22, face = "bold",
                              hjust = 0.5,
                              margin = margin(b = 15)),
    axis.line = element_line(color = "black"),
    axis.title = element_text(size = 22, color = "black",
                              face = "bold"),
    axis.text = element_text(size = 22, color = "black"),
    axis.text.x = element_text(size = 19,margin = margin(t = 5), family = new),
    axis.text.y = element_text(size = 19, margin = margin(r = 5), family = new),
    axis.title.y = element_text(margin = margin(r = 10)),
  axis.title.x = element_text(margin = margin(t = 10)),
    legend.position = c(0.9, 0.95),
    legend.background = element_rect(color = "black"),
```

```
    legend.text = element_text(size = 18),
    legend.margin = margin(t = 5, l = 5, r = 5, b = 5),
    legend.title = element_text(size = 20),
    legend.key = element_rect(color = NA, fill = NA)
) +
  guides(
    fill = guide_legend(
      keywidth = 1.2,
      keyheight = 1.2,
      default.unit = "cm"
    )
  )
```




```
# ggsave("simple_slope.png", width = 10, height = 7, dpi = 1000)
```

6 R 方占比

```
simple_m <- "
CSES_t ~ SNS_t
CSES_t ~ center_SES
CSES_t ~ inter_raw
CSES_t ~ male
CSES_t ~ public
CSES_t ~ freshman
CSES_t ~ sophomore
CSES_t ~ junior
"
fit_simple <- sem(simple_m , data = df2,
                    se = "bootstrap",bootstrap = 5000)

parameterEstimates(fit_simple, standardized = TRUE,
                  rsquare = T, output = "text", header = TRUE)
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws             5000
##   Number of successful bootstrap draws            5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   CSES_t ~
##     SNS_t             1.704    0.174    9.766    0.000    1.365    2.040
##     center_SES        0.468    0.030   15.695    0.000    0.409    0.525
##     inter_raw        -0.073    0.036   -2.040    0.041   -0.145   -0.005
##     male              1.107    0.247    4.485    0.000    0.610    1.594
##     public           -0.022    0.262   -0.084    0.933   -0.535    0.500
```

```
##     freshman         -0.602    0.434   -1.384    0.166   -1.469    0.251
##     sophomore        -0.182    0.429   -0.425    0.670   -1.017    0.667
##     junior           -0.108    0.471   -0.230    0.818   -1.053    0.809
##    Std.lv  Std.all
##
##     1.704    0.289
##     0.468    0.445
##    -0.073   -0.062
##     1.107    0.118
##    -0.022   -0.002
##    -0.602   -0.062
##    -0.182   -0.019
##    -0.108   -0.008
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .CSES_t           14.362    0.656   21.899    0.000   12.959   15.516
##    Std.lv  Std.all
##    14.362    0.656
##
## R-Square:
##                    Estimate
##     CSES_t            0.344
```

```
simple_m_n <- "
CSES_t ~ SNS_t
CSES_t ~ center_SES
CSES_t ~ male
CSES_t ~ public
CSES_t ~ freshman
CSES_t ~ sophomore
CSES_t ~ junior
"
fit_simple_n <- sem(simple_m_n , data = df2,
```

```
                    se = "bootstrap",bootstrap = 5000)

parameterEstimates(fit_simple_n, standardized = TRUE,
                   rsquare = T, output = "text", header = TRUE)
```

```
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws              5000
##   Number of successful bootstrap draws             5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##   CSES_t ~
##     SNS_t             1.721    0.182    9.474    0.000    1.368    2.080
##     center_SES        0.465    0.029   15.862    0.000    0.407    0.521
##     male              1.129    0.245    4.611    0.000    0.638    1.603
##     public           -0.015    0.262   -0.059    0.953   -0.544    0.481
##     freshman         -0.563    0.445   -1.266    0.206   -1.442    0.275
##     sophomore        -0.160    0.436   -0.368    0.713   -1.015    0.709
##     junior           -0.122    0.478   -0.255    0.799   -1.082    0.816
##    Std.lv  Std.all
##
##     1.721    0.291
##     0.465    0.443
##     1.129    0.120
##    -0.015   -0.002
##    -0.563   -0.058
##    -0.160   -0.016
##    -0.122   -0.009
##
## Variances:
```

```
##                    Estimate  Std.Err  z-value  P(>|z|) ci.lower ci.upper
##    .CSES_t           14.445    0.662   21.835    0.000   13.025   15.686
##    Std.lv  Std.all
##    14.445    0.659
##
## R-Squared:
##                    Estimate
##     CSES_t            0.341
```

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

```
## lavaan 0.6-10 ended normally after 1 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                         8
##
##   Number of observations                          1014
##
## Model Test User Model:
##
##   Test statistic                                 0.000
##   Degrees of freedom                                 0
##
## Model Test Baseline Model:
##
##   Test statistic                               422.415
##   Degrees of freedom                                 7
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    1.000
##   Tucker-Lewis Index (TLI)                       1.000
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2792.661
##   Loglikelihood unrestricted model (H1)      -2792.661
##
##   Akaike (AIC)                                5601.322
##   Bayesian (BIC)                              5640.695
##   Sample-size adjusted Bayesian (BIC)         5615.286
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.000
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.000
##   P-value RMSEA <= 0.05                             NA
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Bootstrap
##   Number of requested bootstrap draws              5000
##   Number of successful bootstrap draws             5000
##
## Regressions:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   CSES_t ~
##     SNS_t             1.721    0.182    9.474    0.000    1.721    0.291
##     center_SES        0.465    0.029   15.862    0.000    0.465    0.443
##     male              1.129    0.245    4.611    0.000    1.129    0.120
```

```
##     public           -0.015    0.262   -0.059    0.953   -0.015   -0.002
##     freshman         -0.563    0.445   -1.266    0.206   -0.563   -0.058
##     sophomore        -0.160    0.436   -0.368    0.713   -0.160   -0.016
##     junior           -0.122    0.478   -0.255    0.799   -0.122   -0.009
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .CSES_t           14.445    0.662   21.835    0.000   14.445    0.659
```

```
simple_mn <- lm(CSES_t ~ center_SES + SNS_t + male + freshman +
                        sophomore + junior + public , data = df2)
summary(simple_mn)
```

```
##
## Call:
## lm(formula = CSES_t ~ center_SES + SNS_t + male + freshman +
##     sophomore + junior + public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.3747  -2.4036   0.0346   2.2885  12.7751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.12454    0.40289  47.468  < 2e-16 ***
## center_SES   0.46529    0.02752  16.910  < 2e-16 ***
## SNS_t        1.72051    0.15677  10.975  < 2e-16 ***
## male         1.12934    0.24349   4.638 3.98e-06 ***
## freshman    -0.56261    0.42113  -1.336    0.182
## sophomore   -0.16024    0.40514  -0.396    0.693
## junior      -0.12196    0.46368  -0.263    0.793
## public      -0.01533    0.26602  -0.058    0.954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.816 on 1006 degrees of freedom
## Multiple R-squared:  0.3407, Adjusted R-squared:  0.3361
## F-statistic: 74.27 on 7 and 1006 DF,  p-value: < 2.2e-16
```

```
simple_slope_m2 <- lm(scale(CSES_t) ~ scale(center_SES) + scale(SNS_t) + scale(inter_ra
                      scale(sophomore) + scale(junior) + scale(public) , data = df2)

summary(simple_slope_m2)
```

```
##
## Call:
## lm(formula = scale(CSES_t) ~ scale(center_SES) + scale(SNS_t) +
##     scale(inter_raw) + scale(male) + scale(freshman) + scale(sophomore) +
##     scale(junior) + scale(public), data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.66413 -0.51640 -0.00597  0.50220  2.55612
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.478e-16  2.553e-02   0.000   1.0000
## scale(center_SES)  4.450e-01  2.613e-02  17.028  < 2e-16 ***
## scale(SNS_t)       2.886e-01  2.652e-02  10.882  < 2e-16 ***
## scale(inter_raw)  -6.169e-02  2.564e-02  -2.406   0.0163 *
## scale(male)        1.181e-01  2.592e-02   4.555 5.89e-06 ***
## scale(freshman)   -6.186e-02  4.324e-02  -1.431   0.1528
## scale(sophomore)  -1.876e-02  4.158e-02  -0.451   0.6519
## scale(junior)     -8.227e-03  3.509e-02  -0.234   0.8147
## scale(public)     -2.302e-03  2.788e-02  -0.083   0.9342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8129 on 1005 degrees of freedom
## Multiple R-squared:  0.3445, Adjusted R-squared:  0.3393
## F-statistic: 66.02 on 8 and 1005 DF,  p-value: < 2.2e-16
```