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)
```

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

```r
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

```r
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 %>% dplyr::select(SNS1:SNS6) %>%
cortest.bartlett()
```

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

```
## $chisq
## [1] 3188.472
##
## $p.value
## [1] 0
##
## $df
## [1] 15
```

```
df1 %>% dplyr::select(SNS1:SNS6) %>%
  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
## SNS1 0.79 0.63 0.37   1
## SNS2 0.76 0.58 0.42   1
## SNS3 0.84 0.71 0.29   1
## SNS4 0.80 0.65 0.35   1
## SNS5 0.82 0.68 0.32   1
## SNS6 0.73 0.53 0.47   1
##
##                 PC1
## SS loadings    3.78
## Proportion Var 0.63
##
```

```
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.11
##  with the empirical chi square  349.24  with prob <  9e-70
##
## Fit based upon off diagonal values = 0.96
```

```
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
```

```
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
```

```
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%)
```

4.3 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
```

4.3.1 筛选控制变量

```
t.test(EIB_t ~ sex, df2)# 显著
```

4.3.1.1 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).
```

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

4.3.1.2 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
```

```r
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).
```

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

4.3.1.3 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)
```

4.3.1.4 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    0.030   32.536    0.000    0.907    1.024
##     SNS_t   (cdsh)    0.376    0.171    2.196    0.028    0.049    0.711
##   CSES_t ~
##     SNS_t     (a1)    2.116    0.213    9.945    0.000    1.714    2.544
##   EIB_t ~
##     male              0.492    0.221    2.220    0.026    0.057    0.921
##     public            0.262    0.251    1.044    0.296   -0.222    0.758
##     freshmn          -0.126    0.379   -0.333    0.739   -0.879    0.610
##     sophomr          -0.400    0.375   -1.066    0.287   -1.146    0.343
##     junior           -0.084    0.460   -0.184    0.854   -0.982    0.812
##   CSES_t ~
##     male              1.390    0.279    4.977    0.000    0.845    1.947
##     public            0.073    0.297    0.245    0.806   -0.521    0.646
##     freshmn          -0.784    0.503   -1.559    0.119   -1.740    0.205
##     sophomr          -0.760    0.488   -1.556    0.120   -1.728    0.226
##     junior           -0.028    0.554   -0.051    0.959   -1.105    1.043
##    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    0.750   17.039    0.000   11.218   14.159
##    .CSES_t           18.550    0.809   22.921    0.000   16.861   20.125
##    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    0.226    9.053    0.000    1.618    2.520
##     direct            0.967    0.030   32.532    0.000    0.907    1.024
##     total             2.421    0.279    8.674    0.000    1.881    2.975
##    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)
```

```r
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    0.283    8.565    0.000
##     male              1.836    0.348    5.273    0.000
##     freshman         -0.884    0.627   -1.409    0.159
##     sophomore        -1.134    0.620   -1.830    0.067
##     junior           -0.112    0.698   -0.160    0.873
##     public            0.332    0.384    0.865    0.387
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
## .EIB_t 30.123 1.356 22.210 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    0.029   33.205    0.000    0.908    1.023
##     SNS_t   (cdsh)    0.376    0.170    2.209    0.027    0.038    0.707
##     male              0.492    0.225    2.186    0.029    0.049    0.929
##     public            0.262    0.253    1.036    0.300   -0.234    0.761
##     freshmn          -0.126    0.384   -0.328    0.743   -0.874    0.617
##     sophomr          -0.400    0.372   -1.075    0.282   -1.127    0.316
##     junior           -0.084    0.451   -0.188    0.851   -0.957    0.798
##   CSES_t ~
##     SNS_t     (a1)    1.704    0.178    9.593    0.000    1.363    2.055
##     cnt_SES   (a2)    0.468    0.030   15.622    0.000    0.408    0.526
##     intr_rw   (a3)   -0.073    0.036   -2.008    0.045   -0.147   -0.005
##     male              1.107    0.242    4.569    0.000    0.635    1.589
```

```
##     public            -0.022    0.263   -0.083    0.934   -0.547    0.497
##     freshmn           -0.602    0.432   -1.392    0.164   -1.467    0.231
##     sophomr           -0.182    0.428   -0.427    0.670   -1.033    0.635
##     junior            -0.108    0.472   -0.230    0.818   -1.040    0.817
##    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    0.757   16.881    0.000   11.230   14.198
##    .CSES_t           14.362    0.651   22.070    0.000   12.942   15.481
##    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    0.260    7.533    0.000    1.468    2.474
##     ind_med           1.647    0.185    8.888    0.000    1.299    2.017
##     ind_high          1.334    0.224    5.965    0.000    0.896    1.760
##     dif1             -0.313    0.157   -1.996    0.046   -0.637   -0.021
##     dif2             -0.625    0.313   -1.996    0.046   -1.273   -0.042
##     dif3             -0.313    0.157   -1.996    0.046   -0.637   -0.021
##     imm              -0.070    0.035   -1.996    0.046   -0.143   -0.005
##     direct            0.376    0.170    2.209    0.027    0.038    0.707
##     total_low         2.335    0.314    7.434    0.000    1.731    2.945
##     total_med         2.023    0.243    8.339    0.000    1.550    2.493
##     total_high        1.710    0.261    6.556    0.000    1.191    2.200
##    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(EIB_t ~ SNS_t + CSES_t + male + freshman + sophomore +
                  junior + public, data = df2)
summary(model_2nd)
```

```
##
## Call:
## lm(formula = EIB_t ~ SNS_t + CSES_t + male + freshman + sophomore +
##     junior + public, data = df2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8917  -1.9831   0.3022   2.5329  15.5836
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.62800    0.63046  15.271   <2e-16 ***
## SNS_t        0.37552    0.15599   2.407   0.0163 *
## CSES_t       0.96679    0.02617  36.938   <2e-16 ***
## male         0.49173    0.23148   2.124   0.0339 *
## freshman    -0.12605    0.39652  -0.318   0.7506
## sophomore   -0.39967    0.38020  -1.051   0.2934
## junior      -0.08449    0.43618  -0.194   0.8464
## public       0.26173    0.25022   1.046   0.2958
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.59 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)
```

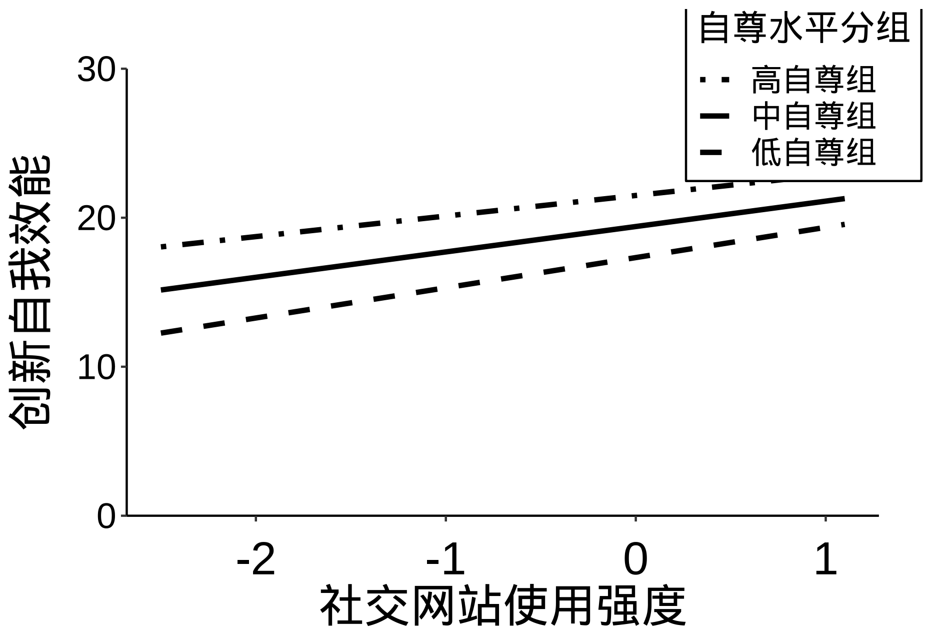
```
## 载入需要的程辑包：carData
```

```
## 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)))

df3 <- as.data.frame(Inter.1a)

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(margin = margin(t = 10)),
    axis.text.y = element_text(size = 17),
    axis.title.y = element_text(margin = margin(r = 10)),
    legend.position = c(0.9, 0.95),
    legend.background = element_rect(color = "black"),
```

```
    legend.text = element_text(size = 15),
    legend.margin = margin(t = 5, l = 5, r = 5, b = 5),
    legend.title = element_text(size = 17),
    legend.key = element_rect(color = NA, fill = NA)
) +
  guides(
    fill = guide_legend(
      keywidth = 1.2,
      keyheight = 1.2,
      default.unit = "cm"
    )
  )
```




```
# ggsave("simple_slop.png", width = 10, height = 7, dpi = 300)
```