

## Load necessary packages

```
library(tidyverse)
library(xlsx)
library(data.table)
library(lavaan)
library(semPlot)
```

## E-CLASS scoring function

```
answers <- read.xlsx('C:/Users/Cole/Documents/ECLASS_DATA/Answers_Template.xlsx', sheetName = 'Converted')

colnames(answers) <- as.character(unlist(answers[2,]))
answers <- answers[1,]
answers <- as.data.frame(t(answers)) %>%
  `colnames<-`(c('Answer')) %>%
  mutate(Question = row.names(.)) %>%
  filter(!grepl('\\\\.1$', Question)) %>%
  mutate(Answer = case_when(
    Answer == 'A' ~ 1,
    Answer == 'D' ~ -1,
    TRUE ~ NA_real_
  )) %>%
  filter(!is.na(Answer))

answers <- data.frame(t(answers)) %>%
  `colnames<-`(as.character(unlist(. [2,]))) %>%
  slice(., 1)

answers.cols <- names(answers)

Read.Score.ECLASS <- function(file){
  dt <- fread(file)

  dt[, (answers.cols) := lapply(.SD, function(x) case_when(x >= 4 ~ 1,
                                                            x <= 2 ~ -1,
                                                            TRUE ~ 0)),
    .SDcols = answers.cols]

  scores.df <- sweep(dt[, ..answers.cols], 2,
                    as.numeric(as.character(unlist(answers))), "*")

  scores.df$student.score <- rowSums(scores.df %>% select(grep("a$", names(.))))
  scores.df$expert.score <- rowSums(scores.df %>% select(grep("b$", names(.))))

  df <- cbind(dt, scores.df[, c('student.score', 'expert.score')]) %>%
    filter((V10 == 1) & (q40a == 4)) %>% # remove students that didn't finish/click filter
    select(Q3_1_TEXT, Q3_2_TEXT, Q3_3_TEXT, student.score, expert.score) %>%
    `colnames<-`(c('first.name', 'last.name', 'ID', 'student.score', 'expert.score'))
```

```

return(df)
}

```

## Read E-CLASS files

```

# Fall 2017

df.ECLASS.P1116.F2017pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/
df.ECLASS.P1116.F2017post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/
df.ECLASS.P2217.F2017pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/
df.ECLASS.P2217.F2017post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/
df.ECLASS.P2218.F2017pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/
df.ECLASS.P2218.F2017post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2017/

# Spring 2018

df.ECLASS.P1116.S2018pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Spring20
df.ECLASS.P1116.S2018post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Spring20

# Fall 2018

df.ECLASS.P1112.F2018pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2018/
df.ECLASS.P1112.F2018post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2018/
df.ECLASS.P1116.F2018pre <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2018/
df.ECLASS.P1116.F2018post <- Read.Score.ECLASS('C:/Users/Cole/Documents/ECLASS_DATA/Raw_Surveys/Fall2018/

```

## Read PLIC files

```

Get.PLIC.data <- function(df, ClassID){
  df.class <- df %>%
    filter(Class_ID == ClassID) %>%
    select(-Class_ID) %>%
    `colnames<-`(c('ID', 'last.name', 'first.name', 'PLIC.score'))

  return(df.class)
}

### Pre ###

```

```

df.PLIC.pre <- fread('C:/Users/Cole/Documents/PLIC_DATA/Collective_Surveys/Complete/Complete_Concat.csv')
  filter(!is.na(PreScores)) %>%
  select(Class_ID, Q5a_x, Q5b_x, Q5c_x, PreScores)

df.PLIC.post <- fread('C:/Users/Cole/Documents/PLIC_DATA/Collective_Surveys/Complete/Complete_Concat.csv')
  filter(!is.na(PostScores)) %>%
  select(Class_ID, Q5a_y, Q5b_y, Q5c_y, PostScores)

# Fall 2017

df.PLIC.P1116.F2017pre <- Get.PLIC.data(df.PLIC.pre, 'R_10ko8BpPfb9rt0G')
df.PLIC.P1116.F2017post <- Get.PLIC.data(df.PLIC.post, 'R_10ko8BpPfb9rt0G')
df.PLIC.P2217.F2017pre <- Get.PLIC.data(df.PLIC.pre, 'R_12QFe4VQPh6oNW1')
df.PLIC.P2217.F2017post <- Get.PLIC.data(df.PLIC.post, 'R_12QFe4VQPh6oNW1')
df.PLIC.P2218.F2017pre <- Get.PLIC.data(df.PLIC.pre, 'R_2Y4jnzAgBixC4Qm')
df.PLIC.P2218.F2017post <- Get.PLIC.data(df.PLIC.post, 'R_2Y4jnzAgBixC4Qm')

# Spring 2018

df.PLIC.P1116.S2018pre <- Get.PLIC.data(df.PLIC.pre, 'R_2R8MnTyv2jFgPzA')
df.PLIC.P1116.S2018post <- Get.PLIC.data(df.PLIC.post, 'R_2R8MnTyv2jFgPzA')

# Fall 2018

df.PLIC.P1112.F2018pre <- Get.PLIC.data(df.PLIC.pre, 'R_3ijRcPfXo8MUfFj')
df.PLIC.P1112.F2018post <- Get.PLIC.data(df.PLIC.post, 'R_3ijRcPfXo8MUfFj')
df.PLIC.P1116.F2018pre <- Get.PLIC.data(df.PLIC.pre, 'R_1IB300CxBKh0Tw7')
df.PLIC.P1116.F2018post <- Get.PLIC.data(df.PLIC.post, 'R_1IB300CxBKh0Tw7')

```

## Matching E-CLASS and PLIC

```

match.ECLASS.PLIC <- function(ECLASS.df, PLIC.df){
  match.cols <- c('ID', 'last.name', 'first.name')
  ECLASS.df[, match.cols] <- sapply(ECLASS.df[, match.cols], tolower)
  PLIC.df[, match.cols] <- sapply(PLIC.df[, match.cols], tolower)

  names.df <- inner_join(ECLASS.df, PLIC.df, by = c('first.name', 'last.name'),
                        suffix = c('', '.copy'))[, c(match.cols, 'student.score',
                                                         'expert.score', 'PLIC.score')]

  ID.df <- inner_join(ECLASS.df, PLIC.df, by = c('ID'),
                     suffix = c('', '.copy'))[, c(match.cols, 'student.score',
                                                         'expert.score', 'PLIC.score')]

  match.df <- rbind(names.df, ID.df) %>%
    filter(!duplicated(ID))
}

P1116.F2017pre.df <- match.ECLASS.PLIC(df.ECLASS.P1116.F2017pre, df.PLIC.P1116.F2017pre)
P2217.F2017pre.df <- match.ECLASS.PLIC(df.ECLASS.P2217.F2017pre, df.PLIC.P2217.F2017pre)
P2218.F2017pre.df <- match.ECLASS.PLIC(df.ECLASS.P2218.F2017pre, df.PLIC.P2218.F2017pre)
P1116.S2018pre.df <- match.ECLASS.PLIC(df.ECLASS.P1116.S2018pre, df.PLIC.P1116.S2018pre)

```

```

P1112.F2018pre.df <- match.ECLASS.PLIC(df.ECLASS.P1112.F2018pre, df.PLIC.P1112.F2018pre)
P1116.F2018pre.df <- match.ECLASS.PLIC(df.ECLASS.P1116.F2018pre, df.PLIC.P1116.F2018pre)

pre.df <- rbind(P1116.F2017pre.df, P2217.F2017pre.df, P2218.F2017pre.df,
                P1116.S2018pre.df, P1112.F2018pre.df, P1116.F2018pre.df)

P1116.F2017post.df <- match.ECLASS.PLIC(df.ECLASS.P1116.F2017post, df.PLIC.P1116.F2017post)
P2217.F2017post.df <- match.ECLASS.PLIC(df.ECLASS.P2217.F2017post, df.PLIC.P2217.F2017post)
P2218.F2017post.df <- match.ECLASS.PLIC(df.ECLASS.P2218.F2017post, df.PLIC.P2218.F2017post)
P1116.S2018post.df <- match.ECLASS.PLIC(df.ECLASS.P1116.S2018post, df.PLIC.P1116.S2018post)
P1112.F2018post.df <- match.ECLASS.PLIC(df.ECLASS.P1112.F2018post, df.PLIC.P1112.F2018post)
P1116.F2018post.df <- match.ECLASS.PLIC(df.ECLASS.P1116.F2018post, df.PLIC.P1116.F2018post)

post.df <- rbind(P1116.F2017post.df, P2217.F2017post.df, P2218.F2017post.df,
                 P1116.S2018post.df, P1112.F2018post.df, P1116.F2018post.df)

```

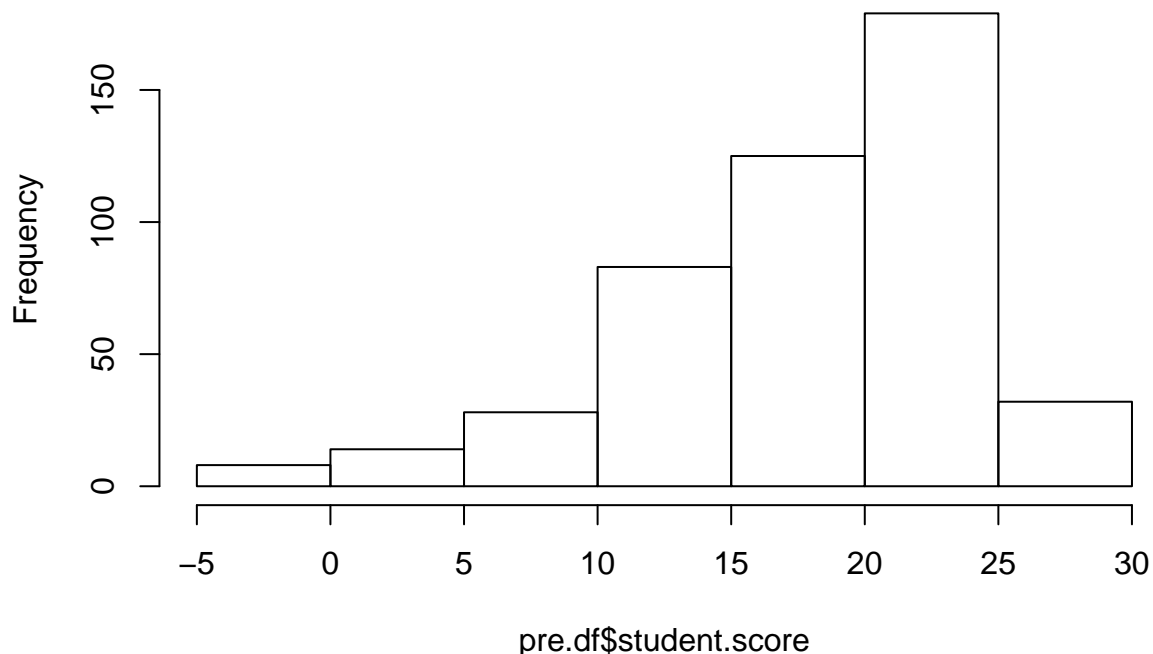
## Correlation and visualization (PRE)

```
nrow(pre.df)
```

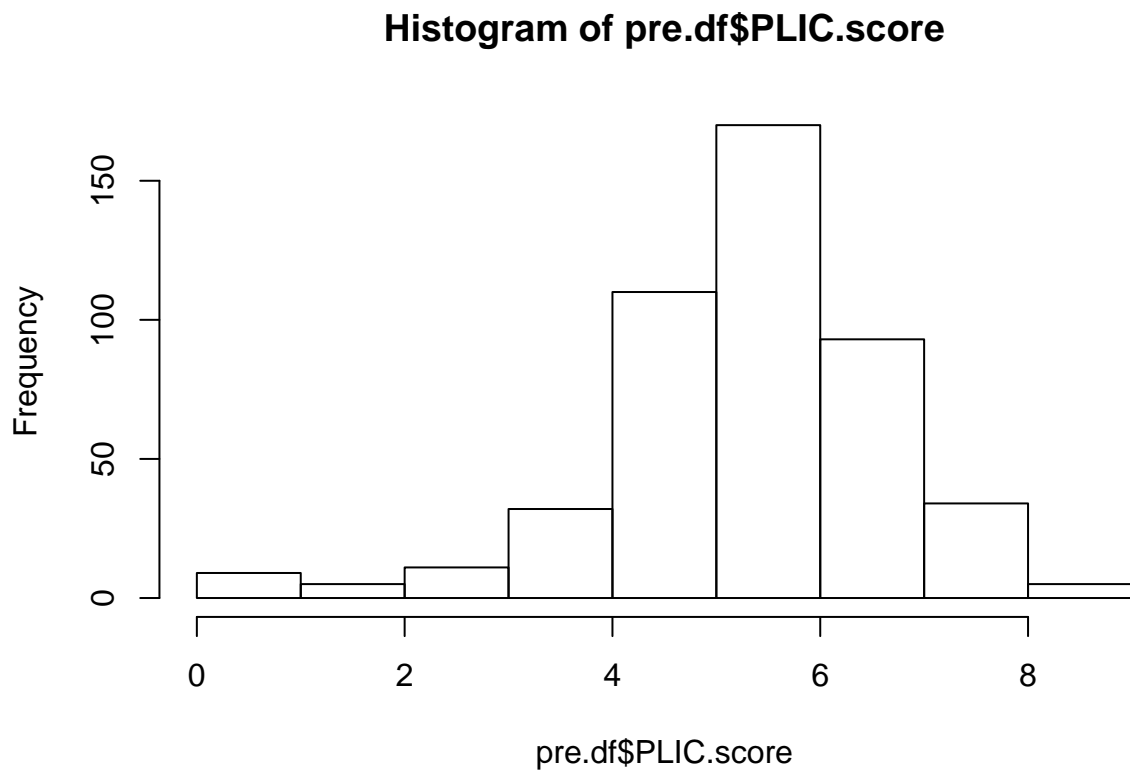
```
## [1] 469
```

```
hist(pre.df$student.score)
```

**Histogram of pre.df\$student.score**



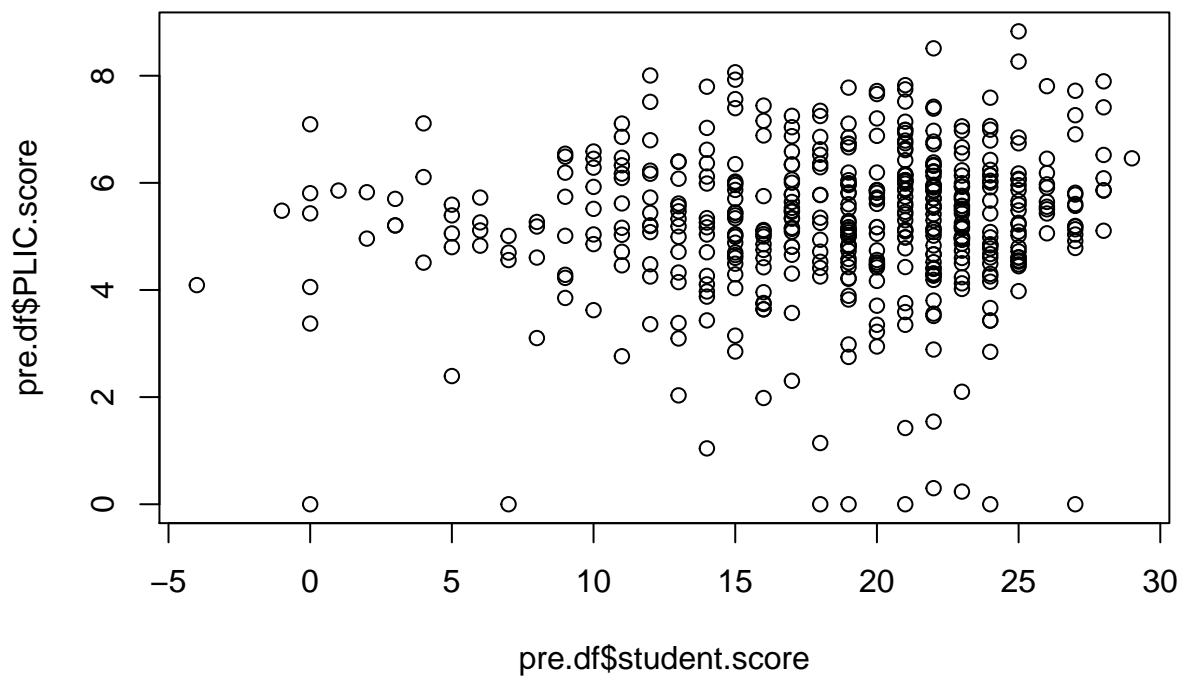
```
hist(pre.df$PLIC.score)
```



```
cor(pre.df$student.score, pre.df$PLIC.score)
```

```
## [1] 0.09612278
```

```
plot(pre.df$student.score, pre.df$PLIC.score)
```



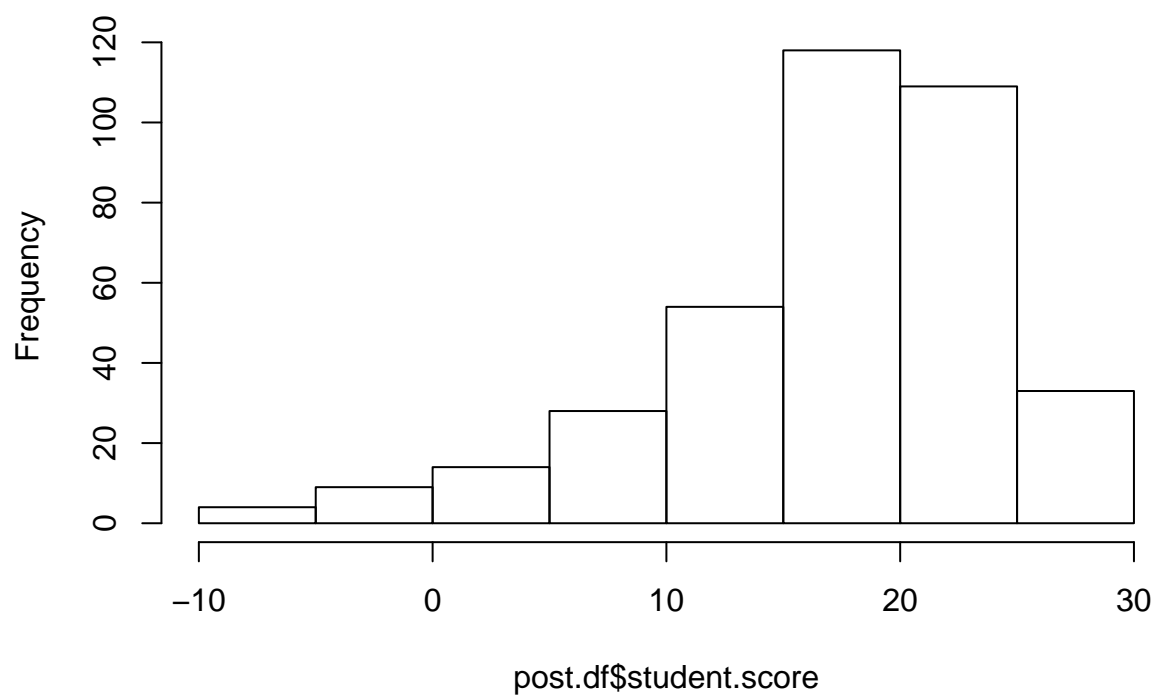
## Correlation and visualization (POST)

```
nrow(post.df)
```

```
## [1] 369
```

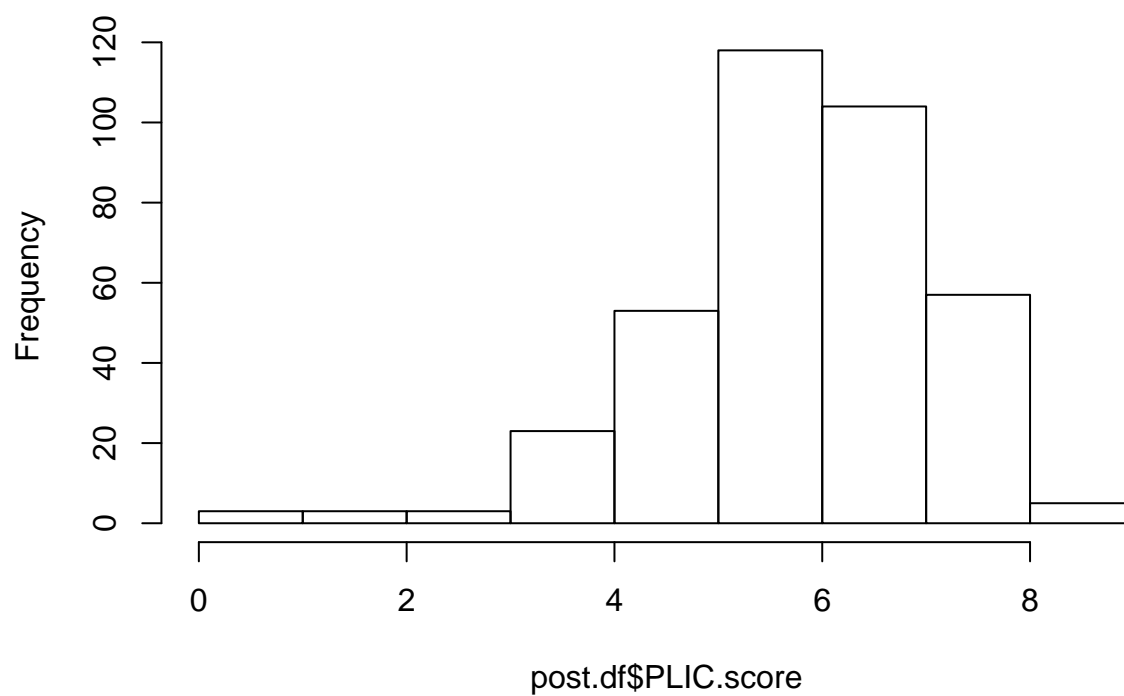
```
hist(post.df$student.score)
```

**Histogram of post.df\$student.score**



```
hist(post.df$PLIC.score)
```

**Histogram of post.df\$PLIC.score**

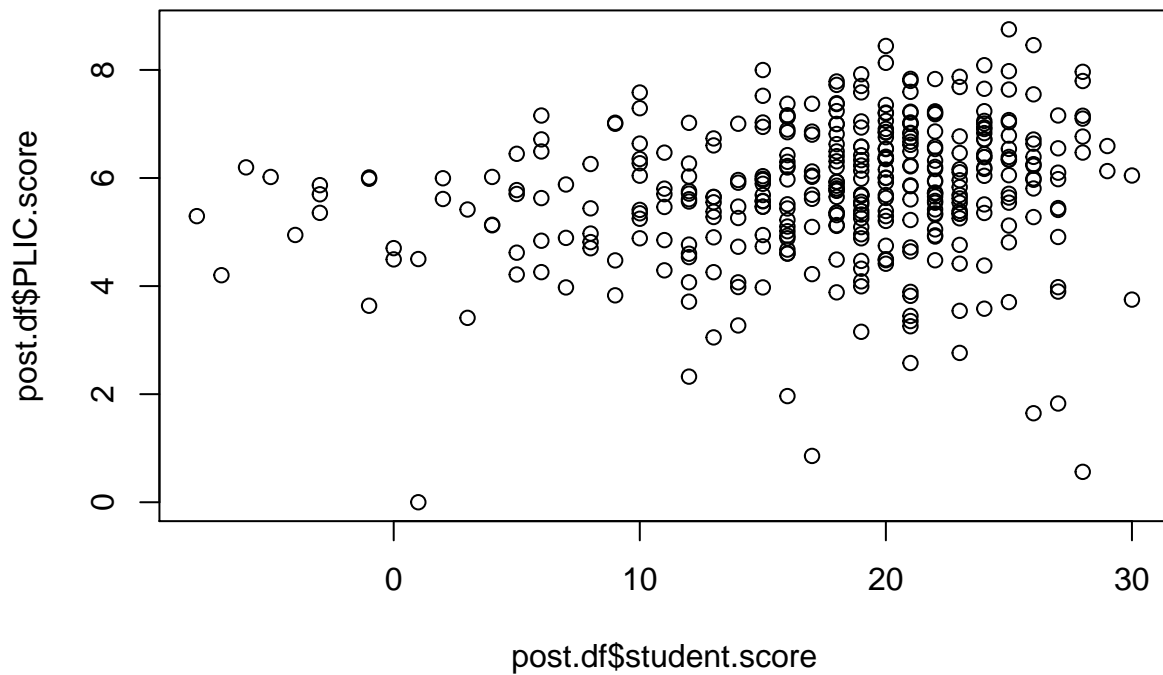


```
cor(post.df$student.score, post.df$PLIC.score)
```

```
## [1] 0.1970626
```

```
plot(post.df$student.score, post.df$PLIC.score)
```





## SEM analysis

```
names.df <- inner_join(pre.df, post.df, by = c('first.name', 'last.name'),
                      suffix = c('.pre', '.post')) %>%
  select(-ID.post)
colnames(names.df)[1] <- 'ID'

ID.df <- inner_join(pre.df, post.df, by = c('ID'),
                  suffix = c('.pre', '.post')) %>%
  select(-c('first.name.post', 'last.name.post'))
colnames(ID.df)[c(2, 3)] <- c('last.name', 'first.name')

df <- rbind(names.df, ID.df) %>%
  filter(!duplicated(ID)) %>%
  select(student.score.pre, student.score.post, PLIC.score.pre, PLIC.score.post)

mod <- '
  PLIC.score.pre ~ student.score.pre
  student.score.post ~ student.score.pre + PLIC.score.pre
  PLIC.score.post ~ PLIC.score.pre + student.score.post
'

fit <- sem(mod, df)
```

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

```
## lavaan 0.6-3 ended normally after 28 iterations
##
## Optimization method          NLMINB
## Number of free parameters      8
##
## Number of observations        337
##
## Estimator                     ML
## Model Fit Test Statistic      0.509
## Degrees of freedom            1
## P-value (Chi-square)         0.476
##
## Model test baseline model:
##
## Minimum Function Test Statistic 231.405
## Degrees of freedom              6
## P-value                        0.000
##
## User model versus baseline model:
##
## Comparative Fit Index (CFI)      1.000
## Tucker-Lewis Index (TLI)        1.013
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0)    -2136.589
## Loglikelihood unrestricted model (H1) -2136.334
##
## Number of free parameters        8
## Akaike (AIC)                    4289.177
## Bayesian (BIC)                  4319.738
## Sample-size adjusted Bayesian (BIC) 4294.361
##
## Root Mean Square Error of Approximation:
##
## RMSEA                          0.000
## 90 Percent Confidence Interval    0.000 0.128
## P-value RMSEA <= 0.05            0.633
##
## Standardized Root Mean Square Residual:
##
## SRMR                          0.009
##
## Parameter Estimates:
##
## Information                    Expected
## Information saturated (h1) model Structured
## Standard Errors                Standard
##
## Regressions:
##
## Estimate Std.Err z-value P(>|z|) Std.lv
```

```

##    PLIC.score.pre ~
##      student.scr.pr      0.007    0.012    0.554    0.580    0.007
##    student.score.post ~
##      student.scr.pr      0.726    0.046   15.860    0.000    0.726
##      PLIC.score.pre      0.396    0.211    1.875    0.061    0.396
##    PLIC.score.post ~
##      PLIC.score.pre      0.245    0.046    5.273    0.000    0.245
##      studnt.scr.pst      0.029    0.009    3.234    0.001    0.029
##    Std.all
##
##      0.030
##
##      0.651
##      0.077
##
##      0.272
##      0.167
##
## Variances:
##              Estimate Std.Err z-value P(>|z|) Std.lv Std.all
##    .PLIC.score.pre    1.814   0.140  12.981   0.000   1.814   0.999
##    .studnt.scr.pst   27.259   2.100  12.981   0.000  27.259   0.567
##    .PLIC.score.pst    1.304   0.100  12.981   0.000   1.304   0.889
##
## Modification Indices:
##
##              lhs op              rhs    mi    epc sepc.lv sepc.all
## 11    PLIC.score.pre ~~    PLIC.score.post 0.508 -2.618 -2.618  -1.702
## 12 student.score.post ~~    PLIC.score.post 0.508 -0.354 -0.354  -0.059
## 14    PLIC.score.pre ~    PLIC.score.post 0.508 -2.008 -2.008  -1.804
## 15 student.score.post ~    PLIC.score.post 0.508 -0.272 -0.272  -0.047
## 16    PLIC.score.post ~ student.score.pre 0.508  0.009  0.009   0.048
## 19 student.score.pre ~    PLIC.score.post 0.508  0.279  0.279   0.054
##    sepc.nox
## 11    -1.702
## 12    -0.059
## 14    -1.804
## 15    -0.047
## 16     0.008
## 19     0.054

```

```
semPaths(fit, whatLabels = 'std', edge.color = 'black', edge.label.cex = 1.5, curve = 2, sizeMan = 12, r
```

