## Load necessary packages

```
library(tidyverse)
library(taRifx)
library(ggrepel)
library(psych)
library(lavaan)
library(semPlot)
library(MVN)
library(infotheo)
source('Calculate-MutInfo.R', local = TRUE)
```

## Create dataframes for analyses

```
# read full dataset of PLIC responses including unmatched responses and course info
df_full <- read.csv('C:/Users/Cole/Documents/PLIC_DATA/Collective_Surveys/Complete/Complete_Concat_Cour

# ideentify columns corresponding to scores on a question and columns corresponding to
# 'other' response choices...we don't care about those since this method evaluates
# utility of response choices and we have to keep the 'other' response regardless
ScoresVec <- c('Q1Bs', 'Q1Ds', 'Q1Es', 'Q2Bs', 'Q2Ds', 'Q2Es', 'Q3Bs', 'Q3Ds', 'Q3Es',
                'Q4Bs')
OthersVec <- c('Q1b_19', 'Q1d_10', 'Q1e_12', 'Q2b_38', 'Q2d_11', 'Q2e_11', 'Q3b_10',
                'Q3d_29', 'Q3e_8', 'Q4b_11')

GetPrePostSurveys <- function(df, survey) {
  # retrieves response choices and scores on either the pre or post survey only
  if(survey == 'PRE') {
    appendix <- 'x' # all presurvey columns have '_x' appended
    df.survey <- df %>%
      # filter only closed response surveys where a presurvey total score exists...the
      # Q3c filter is used to identify most recent versions of the survey where Q3c was
      # included
      filter((Survey_x == 'C') & (!is.na(PreScores)) & (!is.na(Q3c_x)))
  } else {
    appendix <- 'y' # all presurvey columns have '_y' appended
    df.survey <- df %>%
      filter((Survey_y == 'C') & (!is.na(PostScores)) & (!is.na(Q3c_y)))
  }

  # pull all item response choice and score columns, removing the appendix and 'other'
  # response choices
  df.survey <- df.survey %>%
    select(c(grep(paste('((Q1b|Q1d|Q1e|Q2b|Q2d|Q2e|Q3b|Q3d|Q3e|Q4b)_[0-9]*)', appendix,
                        sep = '_'),
                  names(.))), paste(ScoresVec, appendix, sep = '_')) %>%
    `colnames<-`(gsub(x = names(.), pattern = paste("\\", appendix, sep = '_'),
                      replacement = "")) %>%
    select(-OthersVec)
}
```

```r
df_Pre <- GetPrePostSurveys(df_full, 'PRE')
df_Post <- GetPrePostSurveys(df_full, 'POST')

df <- rbind(df_Pre, df_Post) # bind pre and post surveys into one big dataframe

# convert characters to factors
char_vars <- lapply(df, class) == "character"
df[, char_vars] <- lapply(df[, char_vars], as.factor)

# ...and then all factors to numeric
df <- df %>%
  japply(., which(sapply(., class) == 'factor'), function(x) as.numeric(levels(x))[x])

df[is.na(df)] <- 0
df_Questions <- df[, ScoresVec] # get data.frame of scores on questions
df_Items <- df[, !names(df) %in% ScoresVec] # and a data.frame of response choices
```

## CFA on dataset with hypothesized model

```r
# hypothesized factor model
PLIC.model.HYP <- ' models  =~ Q1Bs + Q2Bs + Q3Bs + Q3Ds
              methods =~ Q1Ds + Q2Ds + Q4Bs
              actions =~ Q1Es + Q2Es + Q3Es '

mod.cfa.HYP <- cfa(PLIC.model.HYP, data = df_Questions, std.lv = TRUE, estimator = 'ML')

summary(mod.cfa.HYP, fit.measures = TRUE, modindices = FALSE, standardized = TRUE)
```

```
## lavaan 0.6-3 ended normally after 71 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         23
##
##   Number of observations                         13608
##
##   Estimator                                         ML
##   Model Fit Test Statistic                     666.260
##   Degrees of freedom                                32
##   P-value (Chi-square)                           0.000
##
## Model test baseline model:
##
##   Minimum Function Test Statistic             6801.835
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User model versus baseline model:
##
##   Comparative Fit Index (CFI)                    0.906
##   Tucker-Lewis Index (TLI)                       0.868
```

```
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -1559.012
##   Loglikelihood unrestricted model (H1)      -1225.882
##
##   Number of free parameters                         23
##   Akaike (AIC)                                3164.024
##   Bayesian (BIC)                              3336.947
##   Sample-size adjusted Bayesian (BIC)         3263.855
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.038
##   90 Percent Confidence Interval          0.036  0.041
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.030
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard Errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   models =~
##     Q1Bs              0.099    0.004   23.270    0.000    0.099    0.272
##     Q2Bs              0.145    0.004   38.673    0.000    0.145    0.569
##     Q3Bs              0.146    0.004   39.033    0.000    0.146    0.583
##     Q3Ds              0.038    0.003   12.775    0.000    0.038    0.149
##   methods =~
##     Q1Ds              0.105    0.003   35.831    0.000    0.105    0.475
##     Q2Ds              0.134    0.003   39.978    0.000    0.134    0.600
##     Q4Bs              0.076    0.003   24.205    0.000    0.076    0.290
##   actions =~
##     Q1Es              0.105    0.004   29.494    0.000    0.105    0.411
##     Q2Es              0.081    0.003   29.008    0.000    0.081    0.401
##     Q3Es              0.084    0.003   25.661    0.000    0.084    0.343
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   models ~~
##     methods           0.271    0.017   16.214    0.000    0.271    0.271
##     actions           0.401    0.020   20.210    0.000    0.401    0.401
##   methods ~~
##     actions           0.589    0.021   28.548    0.000    0.589    0.589
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .Q1Bs              0.122    0.002   76.718    0.000    0.122    0.926
##    .Q2Bs             0.044    0.001   41.314    0.000    0.044    0.676
##    .Q3Bs             0.041    0.001   39.214    0.000    0.041    0.661
##    .Q3Ds             0.063    0.001   80.882    0.000    0.063    0.978
##    .Q1Ds             0.038    0.001   56.564    0.000    0.038    0.775
##    .Q2Ds             0.032    0.001   37.000    0.000    0.032    0.639
##    .Q4Bs             0.063    0.001   75.327    0.000    0.063    0.916
##    .Q1Es             0.055    0.001   60.728    0.000    0.055    0.831
##    .Q2Es             0.034    0.001   62.042    0.000    0.034    0.839
##    .Q3Es             0.053    0.001   68.533    0.000    0.053    0.883
##     models           1.000                               1.000    1.000
##     methods          1.000                               1.000    1.000
##     actions          1.000                               1.000    1.000
```

```
resid(mod.cfa.HYP) # residual covariances from model
```

```
## $type
## [1] "raw"
##
## $cov
##      Q1Bs   Q2Bs   Q3Bs   Q3Ds   Q1Ds   Q2Ds   Q4Bs   Q1Es   Q2Es   Q3Es
## Q1Bs  0.000
## Q2Bs -0.001  0.000
## Q3Bs -0.001  0.001  0.000
## Q3Ds  0.002 -0.002 -0.001  0.000
## Q1Ds  0.004  0.001  0.001  0.002  0.000
## Q2Ds  0.004 -0.002 -0.001  0.001  0.000  0.000
## Q4Bs  0.002 -0.001 -0.001  0.000 -0.002  0.002  0.000
## Q1Es -0.001 -0.002 -0.002  0.002  0.004  0.002  0.003  0.000
## Q2Es  0.002  0.001  0.000  0.003 -0.001 -0.002 -0.002 -0.001  0.000
## Q3Es  0.002  0.000  0.000  0.006 -0.001 -0.002 -0.003 -0.002  0.003  0.000
```

```
cor(df_Questions) # correlations of question scores
```

```
##            Q1Bs       Q1Ds       Q1Es       Q2Bs       Q2Ds       Q2Es
## Q1Bs 1.00000000 0.07888368 0.02941511 0.14776036 0.09486065 0.06996730
## Q1Ds 0.07888368 1.00000000 0.19420575 0.09937949 0.27986664 0.09225627
## Q1Es 0.02941511 0.19420575 1.00000000 0.05949493 0.18494568 0.14380885
## Q2Bs 0.14776036 0.09937949 0.05949493 1.00000000 0.06578737 0.10711668
## Q2Ds 0.09486065 0.27986664 0.18494568 0.06578737 1.00000000 0.10486557
## Q2Es 0.06996730 0.09225627 0.14380885 0.10711668 0.10486557 1.00000000
## Q3Bs 0.14773060 0.08652311 0.06118749 0.34250986 0.07356949 0.10189203
## Q3Ds 0.06318287 0.04773406 0.05398905 0.05993464 0.04409765 0.07912041
## Q3Es 0.06015201 0.08005173 0.10770640 0.08169243 0.09075708 0.19855033
## Q4Bs 0.04080177 0.10501896 0.10728560 0.02309506 0.20532651 0.03461808
##            Q3Bs       Q3Ds       Q3Es       Q4Bs
## Q1Bs 0.14773060 0.06318287 0.06015201 0.04080177
## Q1Ds 0.08652311 0.04773406 0.08005173 0.10501896
## Q1Es 0.06118749 0.05398905 0.10770640 0.10728560
## Q2Bs 0.34250986 0.05993464 0.08169243 0.02309506
## Q2Ds 0.07356949 0.04409765 0.09075708 0.20532651
## Q2Es 0.10189203 0.07912041 0.19855033 0.03461808
## Q3Bs 1.00000000 0.07686805 0.08796894 0.02942174
## Q3Ds 0.07686805 1.00000000 0.11431078 0.01879198
## Q3Es 0.08796894 0.11431078 1.00000000 0.01236730
## Q4Bs 0.02942174 0.01879198 0.01236730 1.00000000
```

```
semPaths(mod.cfa.HYP, what = 'diagram', whatLabels = 'stand', layout = 'tree2',
         residuals = FALSE, nCharNodes = 10, edge.color = 'black', edge.label.cex = 2,
         curve = 2, label.scale = FALSE, nodeLabels = c('Q1B', 'Q2B', 'Q3B', 'Q3D',
                                                      'Q1D', 'Q2D', 'Q4B', 'Q1E',
                                                      'Q2E', 'Q3E', 'Evaluate\nModels',
                                                      'Evaluate\nMethods',
                                                      'Suggest\nFollow-ups'),
         rotation = 2, sizeMan = 8, sizeLat = 18, width = 4, height = 5, filetype = 'png',
         filename = 'Figures/CFA', mar = c(1, 6, 1, 2))
```

```
## Output stored in C:/Users/Cole/Documents/GitHub/PLIC/MutualInformation/Figures/CFA.png
```

## Calculate and discretize factor scores

```
scores.df <- data.frame(lavPredict(mod.cfa.HYP)) # get factor scores

# calculate optimal number of bins to discretize each of the factor scores
N_models <- floor((max(scores.df$models) - min(scores.df$models))/
                    (3.5 * sd(scores.df$models)/(nrow(scores.df)^(1/3))))
N_methods <- floor((max(scores.df$methods) - min(scores.df$methods))/
                     (3.5 * sd(scores.df$methods)/(nrow(scores.df)^(1/3))))
N_actions <- floor((max(scores.df$actions) - min(scores.df$actions))/
                     (3.5 * sd(scores.df$actions)/(nrow(scores.df)^(1/3))))

# discretize each of the factor score columns and store in a data.frame scores
scores <- data.frame(discretize(scores.df$models, nbins = N_models), discretize(scores.df$methods, nbins
colnames(scores) <- c('models', 'methods', 'actions')
```

## Mutual information between item response choices and individual factors

### Mutual information for item response choices with models factor

```
# get item response choices corresponding to questions included in CFA for the models
# factor
Models.df <- df_Items[, grep('Q1b|Q2b|Q3b|Q3d', names(df_Items))]

# calculate mutual information between response choices and model factor scores with
# bootstrap confidence intervals
Models.MI.df <- MI.CI(Models.df, scores$models, reps = 100)

# we highlight these specific response choices for reasons discussed in text
labels.list <- c('Q2B_9', 'Q2B_21', 'Q3B_9', 'Q3B_21', 'Q2B_11', 'Q3B_11', 'Q2B_8', 'Q3B_23')
```

```r
# plot mutual information with factor versus proportion of times selected for each
# response choice
png('Figures/MutInfo_Models.png')
ggplot(Models.MI.df, aes(x = Prop.Sel, y = MI, color = Question, shape = Question)) +
  geom_point(size = 3.5, alpha = 0.25) +
  geom_errorbar(aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 1,
                alpha = 0.25) +
  geom_point(data = Models.MI.df[Models.MI.df[, 'Item'] %in% labels.list,],
             aes(x = Prop.Sel, y = MI, color = Question), size = 3.5) +
  geom_errorbar(data = Models.MI.df[Models.MI.df[, 'Item'] %in% labels.list,],
                aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 0.8, alpha = 1) +
  scale_shape_manual(values = c(15, 16, 17, 18)) +
  scale_color_manual(values = c("#0072b2", "#d55e00", "#009e73", "#009e73")) +
  scale_fill_manual(values = labels.list) +
  geom_text_repel(data = subset(Models.MI.df, Item %in% labels.list),
                  aes(x = Prop.Sel, y = MI, color = Question, label = Item),
                  nudge_x = 0.05, nudge_y = 0.03, size = 6) +
  theme_classic() +
  theme(text = element_text(size = 18)) +
  labs(x = 'Fraction of times selected', y = 'Mutual information (bits)') +
  ylim(0, 0.31)
dev.off()
```

```
## pdf
##   2
```

### Mutual information for item response choices with methods factor

```r
# get item response choices corresponding to questions included in CFA for the methods
# factor
Methods.df <- df_Items[, grep('Q1d|Q2d|Q4b', names(df_Items))]

Methods.MI.df <- MI.CI(Methods.df, scores$methods, reps = 100)

labels.list <- c('Q1D_61', 'Q1D_63', 'Q2D_35', 'Q2D_4', 'Q1D_3', 'Q2D_33', 'Q4B_4')

png('Figures/MutInfo_Methods.png')
ggplot(Methods.MI.df, aes(x = Prop.Sel, y = MI, color = Question, shape = Question)) +
  geom_point(size = 3.5, alpha = 0.25) +
  geom_errorbar(aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 1,
                alpha = 0.25) +
  geom_point(data = Methods.MI.df[Methods.MI.df[, 'Item'] %in% labels.list,],
             aes(x = Prop.Sel, y = MI, color = Question), size = 3.5) +
  geom_errorbar(data = Methods.MI.df[Methods.MI.df[, 'Item'] %in% labels.list,],
                aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 0.8, alpha = 1) +
  scale_shape_manual(values = c(15, 16, 17)) +
  scale_color_manual(values = c("#0072b2", "#d55e00", "#cc79a7")) +
  scale_fill_manual(values = labels.list) +
  geom_text_repel(data = subset(Methods.MI.df, Item %in% labels.list),
            aes(x = Prop.Sel, y = MI, color = Question, label = Item), nudge_x = 0.02,
            nudge_y = 0.06, size = 6) +
  theme_classic() +
  theme(text = element_text(size = 18)) +
  labs(x = 'Fraction of times selected', y = 'Mutual information (bits)')
dev.off()
```

```
## pdf
##   2
```

## Mutual information for item response choices with actions factor

```r
# get item response choices corresponding to questions included in CFA for the actions
# factor
Actions.df <- df_Items[, grep('Q1e|Q2e|Q3e', names(df_Items))]

Actions.MI.df <- MI.CI(Actions.df, scores$actions, reps = 100)

labels.list <- c('Q1E_1', 'Q1E_4', 'Q1E_13', 'Q2E_14', 'Q2E_6', 'Q3E_11', 'Q3E_13',
                 'Q3E_20')

png('Figures/MutInfo_Actions.png')
ggplot(Actions.MI.df, aes(x = Prop.Sel, y = MI, color = Question, shape = Question)) +
  geom_point(size = 3.5, alpha = 0.25) +
  geom_errorbar(aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 1,
                alpha = 0.25) +
  geom_point(data = Actions.MI.df[Actions.MI.df[, 'Item'] %in% labels.list,],
             aes(x = Prop.Sel, y = MI, color = Question), size = 3.5) +
  geom_errorbar(data = Actions.MI.df[Actions.MI.df[, 'Item'] %in% labels.list,],
                aes(ymin = CI.Low, ymax = CI.High), width = 0.01, size = 0.8, alpha = 1) +
  scale_shape_manual(values = c(15, 16, 17)) +
  scale_color_manual(values = c("#0072b2", "#d55e00", "#009e73")) +
  scale_fill_manual(values = labels.list) +
  geom_text_repel(data = subset(Actions.MI.df, Item %in% labels.list),
            aes(x = Prop.Sel, y = MI, color = Question, label = Item), nudge_x = 0.04,
            nudge_y = 0.01, size = 6) +
  theme_classic() +
  theme(text = element_text(size = 18)) +
  labs(x = 'Fraction of times selected', y = 'Mutual information (bits)')
dev.off()
```

```
## pdf
##   2
```