Survey Data Management, Reliability, and PCA

A Case Study on Self-compasssion and Gratitude Survey

Cheng Peng

Contents

1 Introduction

The goal of this research project is to measure the level of self-compassion as well as the self-care of BSW and MSW students in a Social Work Program at a regional University. We will be using the following two reliable and validated instruments to measure their level of self-compassion and self-care as they immerse themselves in the helping profession. We hope to see how the SC of the students correlates to other independent variables i.e. undergrad/grad program social work, age, education level, religiosity, spirituality, gender, etc.

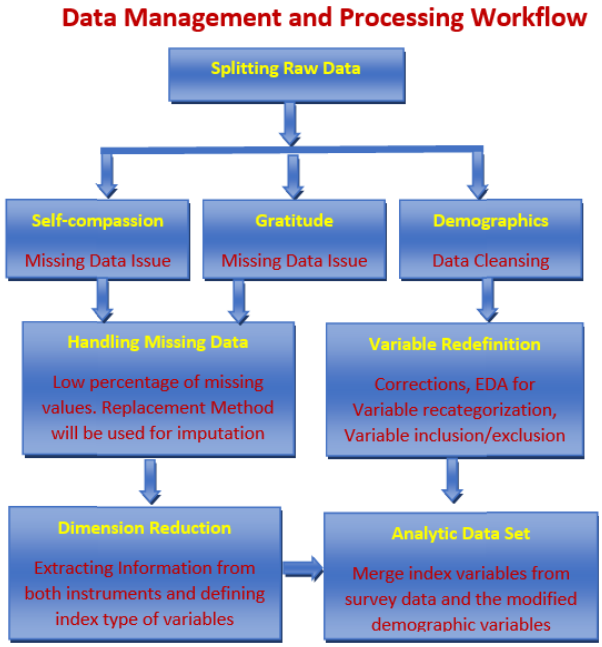
1. The Self-Compassion Scale
2. The Gratitude Questionnaire

The purpose of our research is to study the perception of self-compassion in social work students and how it can link to self-care as well as success in the social work program and field. This will help provide students with self-care practices during their training to thrive in the profession in the future.

2 Data Management and Analyzing Survey Instruments

The next chart illustrates the data management workflow to create the analytic data set.

```
include_graphics("img/w11-DataManagementWorkFlow.png")
```




```
survey = read.csv("w12-SurveyDataCsvFinal.csv", head = TRUE)
# names(survey)
```

The original survey data have three components, a self-compassion scale, gratitude questionnaire instruments, and some demographic questions.

The three components have different portions of missing values. We split the original data set into three subsets of data and imputed the missing values related to the self-compassion and gratitude data based on the survey instruments. Since there are only a few missing values, we replace the missing values in each survey question with the mode of the associated survey item. We create indexes of the two instruments separately to aggregate the information in the two survey data sets.

Since R does not have a function to find the model of a given data set, I write the following function to find the model of a data set.

We will perform both principal component analysis (PCA) and exploratory factor analysis (EFA).

```
my.mode = function(dataset){
  freq.tbl = table(dataset)
  max.freq.id=which(freq.tbl==max(freq.tbl))
  mode=names(freq.tbl[max.freq.id])
  as.numeric(mode)
}
```

2.1 Handling Missing Values Self-Compassion Instrument

This instrument contains only the data associated with the 12 items in the survey instrument. In the original data file, the 12 variables are named Q2_1, Q2_2, . . . , Q2_12. We impute the missing value by replacing the missing value in each of the 12 items with the mode of the corresponding survey items. Since there are only a few missing values in this instrument, this imputation will not impact the subsequent PCA and EFA.

```
compassion = survey[, 1:12]
# Imputing with the mode in each survey item
for (i in 1:12) {
  compassion[,i][is.na(compassion[,i])]=my.mode(compassion[,i])
```

```
}
```

2.2 Handling Missing Values in Gratitude Scale

The gratitude questionnaire contains only the variables associated with gratitude questions. The variables used in the original data file are Q3_1, Q3_2, . . . , Q3_6. We use the same mode imputation method to fill in the missing values as used in the above self-compassion survey data. The gratitude questionnaire has even fewer missing values. Any imputation will not impact any subsequent analysis.

```
gratitude = survey[, 13:18]
# Imputing with the mode in each survey item
for (i in 1:6) {
  gratitude[,i][is.na(gratitude[,i])]=my.mode(gratitude[,i])
}
```

Since Likert scales of the Q3_3 and Q3_6 were in reverse order in the design. We transform back the usual order and create a new dataset using the same variable names.

```
gratitude.new = gratitude
gratitude.new$Q3_3 = 8-gratitude$Q3_3
gratitude.new$Q3_6 = 8-gratitude$Q3_6
```

2.3 Handling Demographic Variables

The demographic variables have two issues: missing values and imbalance categories. Since the size of the data set is slightly close to 120, imputing missing values in a meaningful way is crucial to maintaining the sample size and the statistical power of all subsequent association analyses. About 15 records in the data sets do not have demographic information. Therefore these records were deleted from the final data.

A few missing values occurred in the years of education and employment that are imputed using the auxiliary information in the variables of age, the years of education, and the length of employment.

The major issue of these categorical variables is the imbalance category. The following modifications to the original demographic variables are utilized.

```
demographic = survey[, -(1:18)]
demographic00=demographic
# replace missing values with 99.
demographic00[is.na(demographic00)] <- 99
# Create a frequency table for collapsing categories
#list(Q8.1=table(demographic00$Q8_1),
#     Q8.2=table(demographic00$Q8_2),
#     Q8.3=table(demographic00$Q8_3),
#     Q8.5=table(demographic00$Q8_5),
#     Q8.6=table(demographic00$Q8_6),
#     Q.9=table(demographic00$Q9),
#     Q.11=table(demographic00$Q11),
#     Q.13=table(demographic00$Q13),
#     Q.14=table(demographic00$Q14),
#     Q.15=table(demographic00$Q15),
#     Q.16=table(demographic00$Q16),
#     Q.17=table(demographic00$Q17),
#     Q.18=table(demographic00$Q18),
#     Q.19=table(demographic00$Q19),
#     Q.20=table(demographic00$Q20)
#     )
```

```
grp.age = Q8_1:  1 = (3,23], 2 = [24, 30],  3 = [31, 59]
grp.edu = Q8_2:    1 = [0,15] associate, 2 = [15.5,18.5] bachelor, 3 = [19, 25]  advdegree
grp.empl = Q8_3:   1 = [0,5]  entry, 2 = [5.5,10] junior, 3 = [10.5, 35]   senior
kid.num = Q8_5: 1 = (0) No child, 2 = at least one child
home.size = Q8_6:  1 = (1), 2 = (2), 3 = 3 or more

gender = Q9:  1 = (2) female, 2 = (1,3,5,7) male
race = Q11:   1 = (1)   white,   2 = (2,3) other
marital.st = Q13: 1 = (1) single, 2 = (2) married/civil Partner, 3 = other
disability = Q14:  1 = yes,  2 = No
religion = Q15:  1 = 9 (no religion), 2 = (1,2,3,6,7,8,10,11,99) (religion)
Sexual.orient = Q16:  1 = (4)  heterosexual,  2 = (1,2,3,5,6,7,10) other
poli.affil = Q17: 1 = (4)ind, 2 = (5,6,7)  democrats, 3 = (1,2,3, 99) other
SW.Program = Q18: 1 = (1) BSW, 2 = (2,99) MSW
Urbanity = Q19:  1 = (1) urban, 2 = (2) rural, 3 = (3) suburban
Spirituality = Q20: 1 = (1,2,3) low, 2 = (4)   moderate,  3 = (5,6,7) high
```

We re-define the demographic variables based on the above modification. The modified demographic variables will be used in subsequent modeling.

```
Q8.1=demographic00$Q8_1
grp.age=cut(Q8.1, breaks=c(1, 23, 30, 100), labels=c("[1,23]", "[24,30]", "[30,99]"))
#
Q8.2=demographic00$Q8_2
grp.edu=cut(Q8.2, breaks=c(0, 15.5, 19, 100), labels=c("Assoc", "Bachelor", "Adv.deg"))
#
Q8.3=demographic00$Q8_3
grp.empl=cut(Q8.3, breaks=c(-1,5, 9, 100), labels=c("entry", "junior", "senior"))
#
Q8.5=demographic00$Q8_5
kid.num=cut(Q8.5, breaks=c(-1,1,100), labels=c("No-kid", "With-kid"))
#
Q8.6=demographic00$Q8_6
home.size=cut(Q8.6, breaks=c(-1,2,100), labels=c("1-2", "3+"))
#
Q.9=demographic00$Q9
gender=rep("male", length(Q.9))
gender[which(Q.9==2)]="female"
#
Q.11=demographic00$Q11
race=rep("other", length(Q.11))
race[which(Q.11==1)]="white"
#
Q.13=demographic00$Q13
marital.st = rep("other", length(Q.13))
marital.st[which(Q.13==1)]="single"
marital.st[which(Q.13==2)]="married"
#
Q.14=demographic00$Q14
disability=rep("yes", length(Q.14))
disability[which(Q.14==2)]="no"
#
Q.15=demographic00$Q15
religion=rep("religion", length(Q.15))
religion[which(Q.15==1)]="no-religion"
```

```
#
Q.16=demographic00$Q16
sexual.orient=rep("other", length(Q.16))
sexual.orient[which(Q.16==4)]="heterosexual"
#
Q.17=demographic00$Q17
poli.affil = rep("Republican", length(Q.17))
poli.affil[Q.17 %in% c(5,6)]="ModDemocrats"
poli.affil[Q.17 %in% c(7)]="StrongDemocrats"
poli.affil[which(Q.17 ==4)]="independent"
#
Q.18=demographic00$Q18
SW.program=rep("MSW", length(Q.18))
SW.program[which(Q.18==1)]="BSW"
#
Q.19=demographic00$Q19
urbanity=rep("urban", length(Q.19))
urbanity[which(Q.19==2)]="rural"
urbanity[which(Q.19==3)]="suburban"
#
Q.20=demographic00$Q20
spirituality=rep("high", length(Q.20))
spirituality[which(Q.20==2)]="moderate"
spirituality[Q.20 %in% c(1,2,3)]="low"
#
new.demographics=data.frame(grp.age, grp.edu, grp.empl, kid.num, home.size, gender, race,
           marital.st, disability, religion, sexual.orient, poli.affil,
           SW.program, urbanity, spirituality)
#new.demographics
```

3 Validity and Reliability Analyses

This section reports measures of validity and reliability of the two survey instruments.

3.1 Self-compassion Index

We start with some correlation plots to see the relevance of the PCA procedure on the self-compassion data.

```
##
M=cor(compassion)
corrplot.mixed(M, lower.col = "purple", upper = "ellipse", number.cex = .7, tl.cex = 0.7)
```

Figure 1 shows the moderate association between individual survey items. This implies that the PCA is relevant in aggregating the information in the survey items.

Similar to López et. al. (2018), we use the six positive items associated with mindfulness, self-kindness, and the sense of common humanity to define the self-compassion index.

From the pair-wise correlation plot, we can see that some of the items are reversely scored. We next change back the scale so that all items are positively correlated.

```
###
selfcomp = cbind(compassion$Q2_3, compassion$Q2_7, compassion$Q2_10,
                 compassion$Q2_5,compassion$Q2_2,compassion$Q2_6)
###
```

Figure 1: The pairwise correlation plot reveals the potential relevance of PCA. The shape of an ellipse represents the correlation. The skinnier the ellipse, the higher the correlation. The direction reflects whether a correlation is positive or negative. The off-diagonal direction implies s positive correlation while the main diagonal direction implies a negative association.

```
selcold =  cbind(compassion$Q2_1, compassion$Q2_9, compassion$Q2_4,
                 compassion$Q2_8,compassion$Q2_11,compassion$Q2_12)

###  MD = Mindfulness, SK = Self-kindness, CH = sense of common humanity
selfcomp.sub = cbind(MD1=compassion$Q2_3, MD2=compassion$Q2_7,
                     CH1=compassion$Q2_10,CH2=compassion$Q2_5,
                     SK1=compassion$Q2_2,SK2=compassion$Q2_6)

### SJ = self-judement OI = Over-identification, IS = Isolation
selfcold.sub =  cbind(OI1=compassion$Q2_1, OI2=compassion$Q2_9,
                     IS1=compassion$Q2_4, IS2=compassion$Q2_8,
                     SJ1=compassion$Q2_11,SJ2=compassion$Q2_12)
### Simple sum and average
selfcomp.sum = compassion$Q2_3 + compassion$Q2_7 + compassion$Q2_10 +
                 compassion$Q2_5 + compassion$Q2_2 + compassion$Q2_6
selfcomp.avg = selfcomp.sum/6
```

```
##
M=cor(cbind(selfcomp.sub, selfcold.sub))
corrplot.mixed(M, lower.col = "purple", upper = "ellipse", number.cex = .7, tl.cex = 0.7)
```

Next, we make the following heatmap to illustrate the pairwise correlation between the items in the survey instrument based on the positive adjustment of the scale.

```
##
M=cor(selfcomp.sub)
corrplot.mixed(M, lower.col = "purple", upper = "ellipse", number.cex = .7, tl.cex = 0.7)
```

Intern Consistency

With the adjusted scores in the self-compassion instrument, we calculate one of the commonly used internal consistency reliability Cronbach alpha as follows.

Next, we find the Cronbach alpha and its 95% confidence interval.

```
cronbach.sc = as.numeric(alpha(selfcomp.sub)$total[1])
CI.sc = cronbach.alpha.CI(alpha=cronbach.sc, n=104, items=6, conf.level = 0.95)
CI.comp = cbind(LCI = CI.sc[1], alpha = cronbach.sc, UCI =CI.sc[2])
row.names(CI.comp) = ""
kable(CI.comp, caption="Confodence Interval of Cranbach Alpha")
```

Table 1: Confodence Interval of Cranbach Alpha

LCI	alpha	UCI
0.6903734	0.767224	0.8302429

We can see that Cronbach's alpha is 0.77 with a 95% confidence interval (0.70, 0.84) suggesting that the items in the self-compassion instrument have relatively high internal consistency.

3.2 Gratitude Questionaire

In this section, we perform the same analysis on the gratitude survey instrument. First of all, we present a pairwise correlation plot to display the correlation between individual survey items in the gratitude survey instrument.

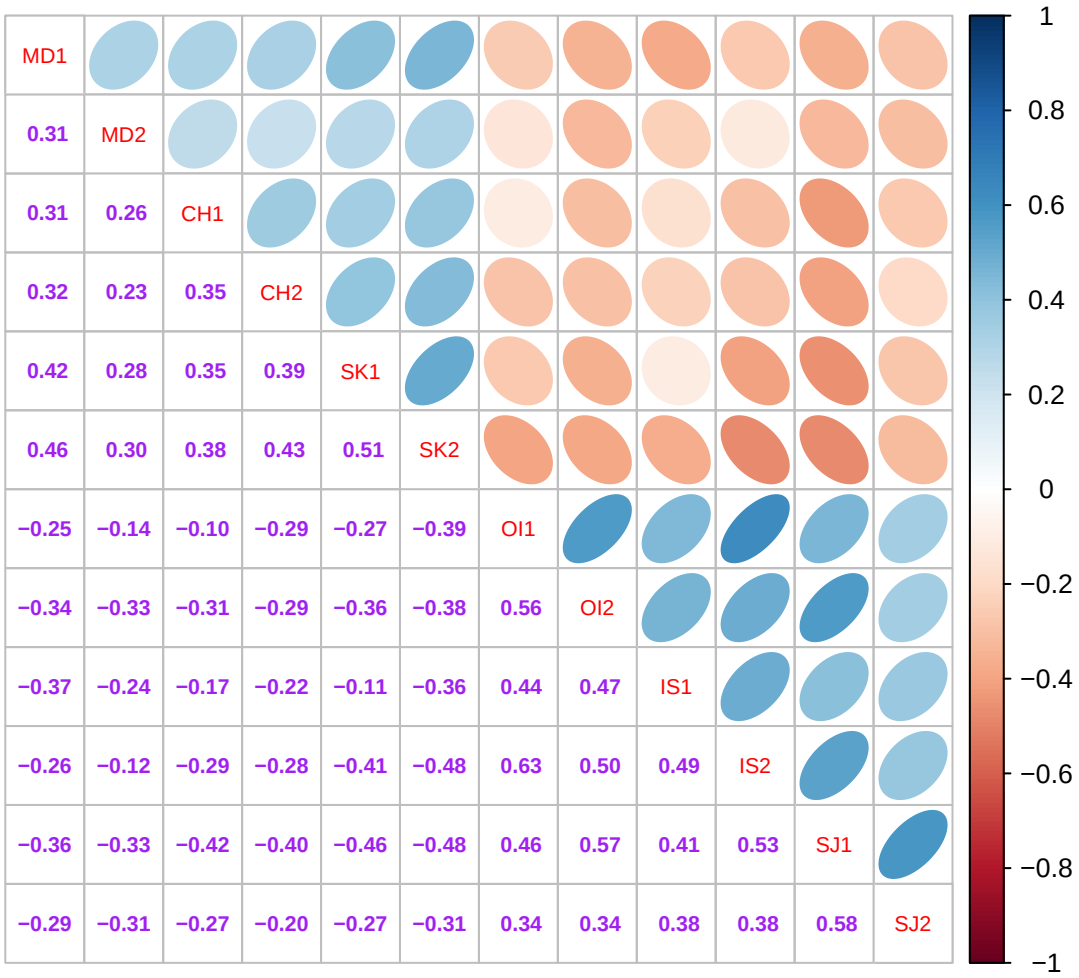


Figure 2: The pairwise correlation plot reveals the potential relevance of PCA. Group items based on self-compassion and self-coldness

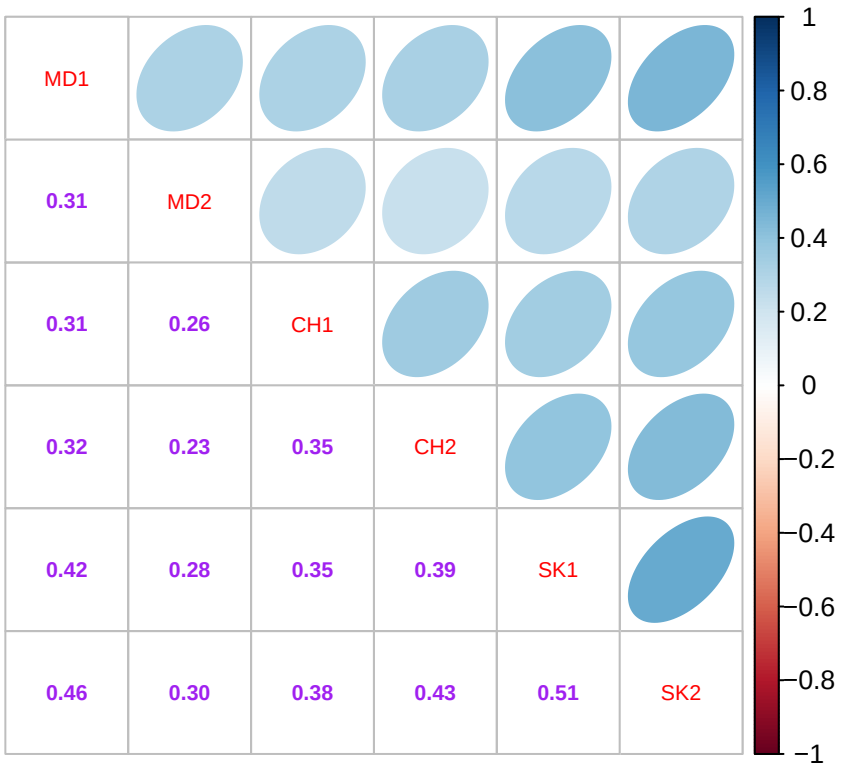


Figure 3: Pairwise correlation based on negatively adjusted subscales. The shape of an ellipse represents the correlation. The skinnier the ellipse, the higher the correlation. The direction reflects whether a correlation is positive or negative. The off-diagonal direction implies s positive correlation while the main diagonal direction implies a negative association.

```
##
M1=cor(gratitude.new)
#corrplot(M, type = "upper", method = "ellipse", main="Pairwise Correlation Plot: Self-Compassion Scale
corrplot.mixed(M1, lower.col = "purple", upper = "ellipse", number.cex = .7, tl.cex = 0.7)
```

Figure 4: The pairwise correlation plot reveals the potential relevance of PCA for the gratitude instrument. The shape of an ellipse represents the correlation. The skinnier the ellipse, the higher the correlation. The direction reflects whether a correlation is positive or negative. The off-diagonal direction implies s positive correlation while the main diagonal direction implies a negative association.

Figure 3 shows that a moderate correlation exists between individual variables. This implies the PCA and EFA can be used to aggregate the information in the set original survey items. Next, we estimate the number of PCAs or EFAs to be retained for the subsequent analysis using the commonly used procedures and summarize the results in the following Figure 4.

Cronbach's Alpha

The internal consistency measure, Cronbach's alpha, of the gratitude instruments calculated below

```
cronbach.gr = as.numeric(alpha(gratitude.new)$total[1])
CI.gr = cronbach.alpha.CI(alpha=cronbach.gr, n=104, items=6, conf.level = 0.95)
CI.gratitude = cbind(LCI = CI.gr[1], alpha = cronbach.gr, UCI =CI.gr[2])
row.names(CI.gratitude) = ""
kable(CI.gratitude, caption="Confodence Interval of Cranbach Alpha")
```

Table 2: Confodence Interval of Cranbach Alpha

LCI	alpha	UCI
0.7280232	0.795529	0.8508849

We can see that Cronbach's alpha is 0.8 with a 95% confidence interval (0.74, 0.86) also suggesting that the items in the Gratitude rating instrument have relatively high internal consistency.

4 PCA Approaches to Information Aggregation

This section reports the results of the principle component analysis. The significant PCs will be added to the analytic data set for regression modeling.

4.1 Two R Functions for PCA

For convenience, we define the following two R functions and use them in the subsequent PCA.

```
My.plotnScree = function(mat, legend = TRUE, method ="factors", main){
    # mat = data matrix
    # method = c("factors", "components"), default is "factors".
    # main = title of the plot
    ev <- eigen(cor(mat))    # get eigenvalues
    ap <- parallel(subject=nrow(mat),var=ncol(mat), rep=5000,cent=.05)
    nScree = nScree(x=ev$values, aparallel=ap$eigen$qevpea, model=method)
    ##
    if (!inherits(nScree, "nScree"))
        stop("Method is only for nScree objects")
    if (nScree$Model == "components")
        nkaiser = "Eigenvalues > mean: n = "
    if (nScree$Model == "factors")
      nkaiser = "Eigenvalues > zero: n = "
    # axis labels
    xlab = nScree$Model
    ylab = "Eigenvalues"
    ##
    par(col = 1, pch = 18)
    par(mfrow = c(1, 1))
    eig <- nScree$Analysis$Eigenvalues
    k <- 1:length(eig)
    plot(1:length(eig), eig, type="b", main = main,
        xlab = xlab, ylab = ylab, ylim=c(0, 1.2*max(eig)))
    #
    nk <- length(eig)
    noc <- nScree$Components$noc
    vp.p <- lm(eig[c(noc + 1, nk)] ~ k[c(noc + 1, nk)])
    x <- sum(c(1, 1) * coef(vp.p))
    y <- sum(c(1, nk) * coef(vp.p))
    par(col = 10)
    lines(k[c(1, nk)], c(x, y))
    par(col = 11, pch = 20)
    lines(1:nk, nScree$Analysis$Par.Analysis, type = "b")
    if (legend == TRUE) {
        leg.txt <- c(paste(nkaiser, nScree$Components$nkaiser),
                    c(paste("Parallel Analysis: n = ", nScree$Components$nparallel)),
                    c(paste("Optimal Coordinates: n = ", nScree$Components$noc)),
                    c(paste("Acceleration Factor: n = ", nScree$Components$naf))
                    )
        legend("topright", legend = leg.txt, pch = c(18, 20, NA, NA),
                           text.col = c(1, 3, 2, 4),
```

```
                            col = c(1, 3, 2, 4), bty="n", cex=0.7)
    }
    naf <- nScree$Components$naf
    text(x = noc, y = eig[noc], label = " (OC)", cex = 0.7,
        adj = c(0, 0), col = 2)
    text(x = naf + 1, y = eig[naf + 1], label = " (AF)",
        cex = 0.7, adj = c(0, 0), col = 4)
}
# example
# My.plotnScree(mat=compassion, legend = TRUE, method ="factors",
#               main = "Number of Factors to Retain")
```

My.loadings.var produces a list of two objects: factor loadings and proportion variance explained by each factor. There are no existing R functions that can be used to extract the proportion of variance from the output of *factanal()*. The function can also extract similar information from the output of a PCA but we need to specify the method in the argument.

```
My.loadings.var <- function(mat, nfct, method="fa"){
   # mat =  data matrix
   # nfct = number of factors or components
   # method = c("fa", "pca"), default = is "fa".
    if(method == "fa"){
     f1 <- factanal(mat, factors = nfct,  rotation = "varimax")
     x <- loadings(f1)
     vx <- colSums(x^2)
     varSS = rbind('SS loadings' = vx,
            'Proportion Var' = vx/nrow(x),
           'Cumulative Var' = cumsum(vx/nrow(x)))
     weight = f1$loadings[]
   } else if (method == "pca"){
     pca <- prcomp(mat, center = TRUE, scale = TRUE)
     varSS = summary(pca)$importance[,1:nfct]
     weight = pca$rotation[,1:nfct]
  }
    list(Loadings = weight, Prop.Var = varSS)
}
# example
# My.loadings.var(mat, nfct=3, method="pca")
```

4.2 PCA Extraction and Number of PCA Determination: Self-compassion

The number of PCAs selected for future exploratory analyses is the key issue and is also the first question we need to address before we move to any further analysis with the PCA scores. Raiche et al (2013) simulation-based test and Scree plot indicate that it is sufficient to choose the first principle component for future analysis. For exploratory purposes, we will choose the first two principal components for both PCA and EFA procedures and use them for association analysis.

```
My.plotnScree(mat=selfcomp.sub, legend = TRUE, method ="components",
              main="Determination of Number of Components\n Self-compassion (Positive)")
```

The above Figure indicates that it is sufficient to retain the first principle component for the subsequent analysis. In the following, we will extract the first two PCAs. The PCA factor loadings and the proportion of variance explained by the retained PCAs are summarized in the following tables.

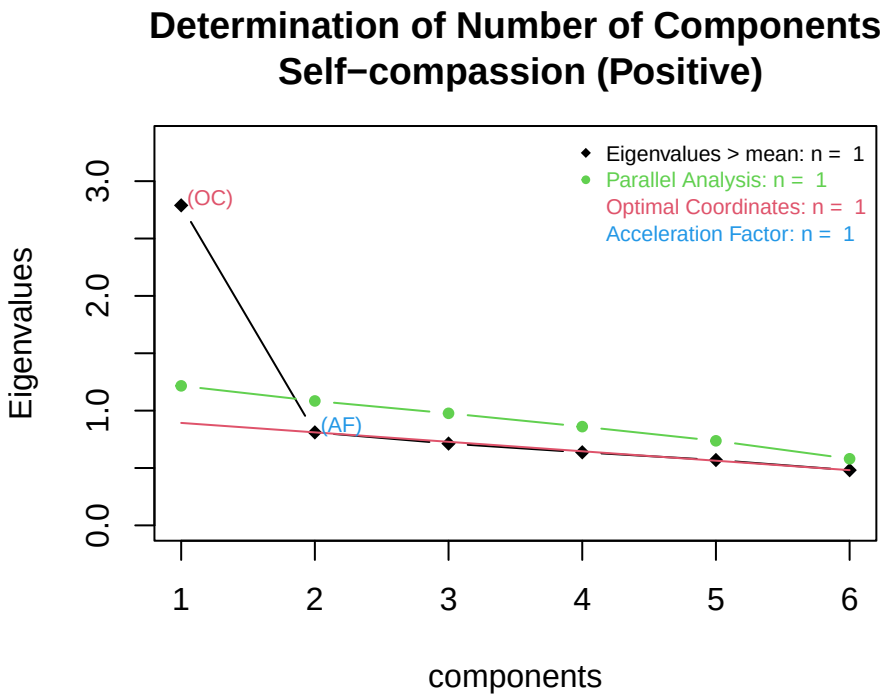


Figure 5: Different methods of identification of the number of principal components to be retained in exploratory analysis: Kaiser's eigenvalue rule, Raiche et al Monte Carlo simulation method (parallel analysis), optimal coordinate (OC) index, and accelerate factor (AF) method.

```
#
Loadings = My.loadings.var(mat=selfcomp.sub, nfct=2, method="pca")$Loadings
#
# pca loadings
kable(round(Loadings,3),
  caption="Factor loadings of the first few PCAs and the cumulative proportion
of variation explained by the corresponding PCAs in the self-compassion survey.")
```

Table 3: Factor loadings of the first few PCAs and the cumulative proportion of variation explained by the corresponding PCAs in the self-compassion survey.

	PC1	PC2
MD1	0.418	0.158
MD2	0.327	0.867
CH1	0.384	-0.156
CH2	0.400	-0.396
SK1	0.442	-0.161
SK2	0.465	-0.129

```
VarProp = My.loadings.var(mat=selfcomp.sub, nfct=2, method="pca")$Prop.Var
# pca loadings
kable(round(VarProp,3),
    caption="Cumulative and proportion of variances explained by each
    the principal component in the self-compassion survey.")
```

Table 4: Cumulative and proportion of variances explained by each the principal component in the self-compassion survey.

	PC1	PC2
Standard deviation	1.670	0.900
Proportion of Variance	0.465	0.135
Cumulative Proportion	0.465	0.600

We also conduct the same analysis using EFA. The Scree type of test also suggests retaining a single factor. The proportion of total variation is lower than that of PCA. we decide to use the PCA method and extract the first two principal components for future analysis. Table 1 shows the factor loadings of the first two principal components. We can see that each of the original items contributes to the two PCAs evenly in terms of the magnitude. The first PCA counts about 41.3% of the total variation and the second PCA counts 10.9% of the total variation. We can simply call the first PCA as *self-compassion index*, denoted by sc.idx.

Next, we extract the self-compassion index in the following code

```
pca <- prcomp(selfcomp, center = TRUE, scale = TRUE)
sc.idx = pca$x[,1]
# hist(sc.idx, breaks=10, main="Distribution of Self-compassion Index")
##
hist(sc.idx,
main="Distribution of Self-compassion Index",
breaks = seq(min(sc.idx), max(sc.idx), length=9),
xlab="Self-compassion Index",
xlim=range(sc.idx),
```

```
border="red",
col="lightblue",
freq=FALSE
)
```

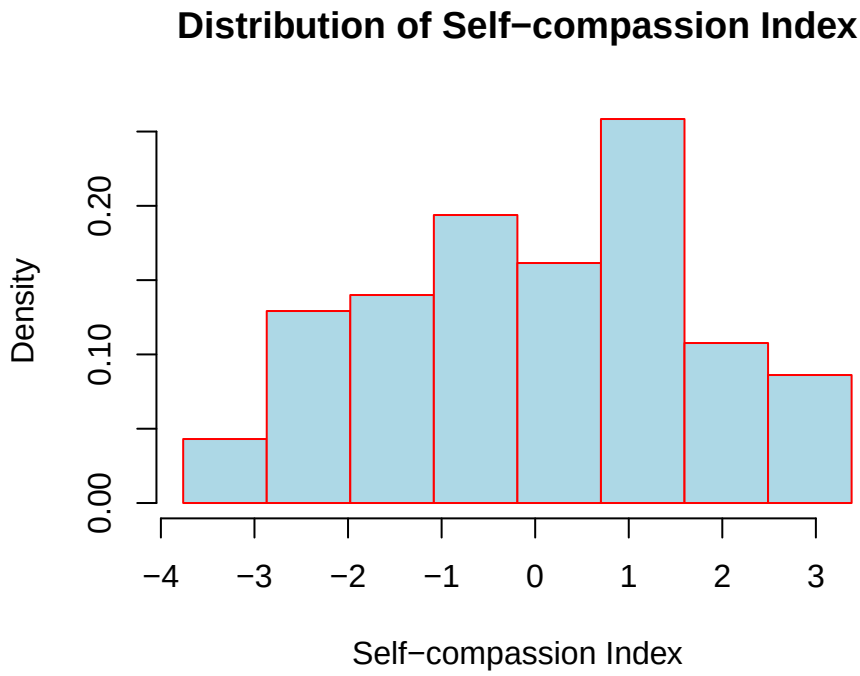


Figure 6: Histogram of the first principle component extract from the self-compassion survey.

4.3 PCA Extraction and Number of PAC Determination: Gratitude

```
My.plotnScree(mat=gratitude.new, legend = TRUE, method ="components",
              main="Determination of Number of Components\n Gratitude Questionnaires")
```

The above Figure indicates retaining one PCA is sufficient for future exploratory analyses. As we did in the self-compassion survey instrument, we extracted the first two principal components for potential analysis. The factor loadings of the two principal components and the corresponding proportion of variation of each component are summarized in the following two tables.

```
Loadings = My.loadings.var(mat=gratitude.new, nfct=2, method="pca")$Loadings
# pca loadings
kable(round(Loadings,3),
    caption="Factor loadings of the first few PCAs and the cumulative
    the proportion of variation explained by the corresponding PCAs in the
    Gratitude Questionaire Survey.")
```

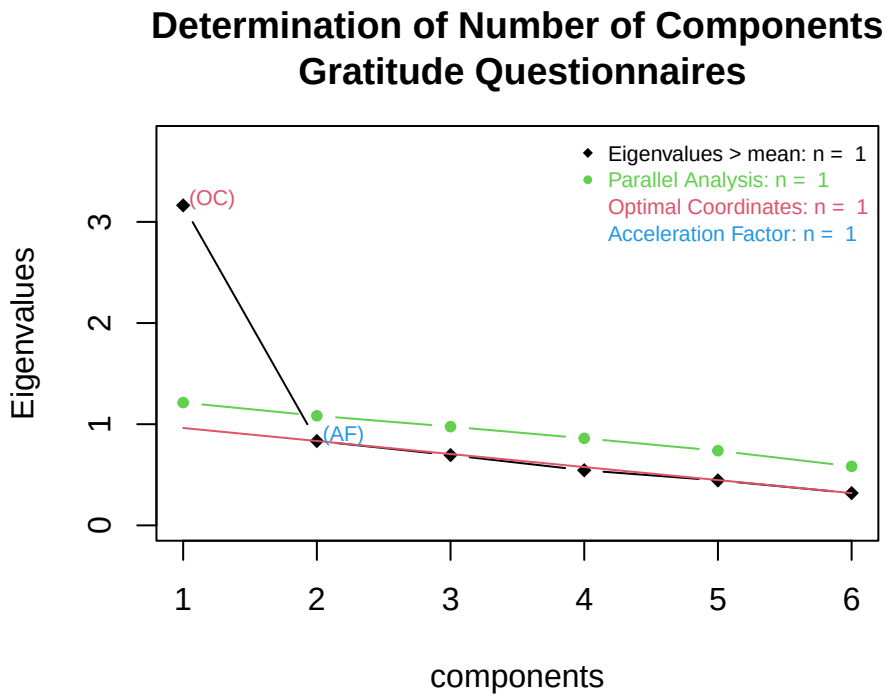


Figure 7: Different methods of identification of the number of principal components to be retained in exploratory analysis for the gratitude survey instrument: Kaiser's eigenvalue rule, Raiche et al Monte Carlo simulation method (parallel analysis), optimal coordinate (OC) index, and accelerate factor (AF) method.

Table 5: Factor loadings of the first few PCAs and the cumulative the proportion of variation explained by the corresponding PCAs in the Gratitude Questionaire Survey.

	PC1	PC2
Q3_1	0.444	0.209
Q3_2	0.462	0.116
Q3_3	0.383	-0.555
Q3_4	0.409	0.236
Q3_5	0.386	0.471
Q3_6	0.356	-0.597

```
VarProp = My.loadings.var(mat=gratitude.new, nfct=6, method="pca")$Prop.Var
# pca loadings
kable(round(VarProp,3),
    caption="Cumulative and proportion of variances explained by each
    principle component from the Gratitude Questionaire Survey.")
```

Table 6: Cumulative and proportion of variances explained by each principle component from the Gratitude Questionaire Survey.

	PC1	PC2	PC3	PC4	PC5	PC6
Standard deviation	1.779	0.913	0.834	0.738	0.667	0.565
Proportion of Variance	0.527	0.139	0.116	0.091	0.074	0.053
Cumulative Proportion	0.527	0.666	0.782	0.873	0.947	1.000

A similar analysis was also conducted using EFA. The result indicates that the first principle factor is sufficient for exploratory analysis. The proportion of the total variation explained by the first factor is about 40% which is about 10% less than that in the first principle component. We will use the first PCA and call it as gratitude index.

Next, we extract the first PCA scores.

```
gr.pca <- prcomp(gratitude.new, center = TRUE, scale = TRUE)
gr.idx = gr.pca$x[,1]
###
hist(gr.idx,
main="Untransformed Gratitude Indx",
breaks = seq(min(gr.idx), max(gr.idx), length=10),
xlab="Gratitude Index",
xlim=range(gr.idx),
border="red",
col="lightblue",
freq=FALSE
)
```

To define a meaningful index of self-compassion, we want to make sure that the proposed index is positively correlated to the individual item. Since the principle component analysis algorithm is essentially an orthogonal rotation (transformation), we can adjust the direction of the coordinate system to make the PCA scores a meaningful index for subsequent association analysis. The is the plot of the pairwise association between the individual items and the two new PCA scores.

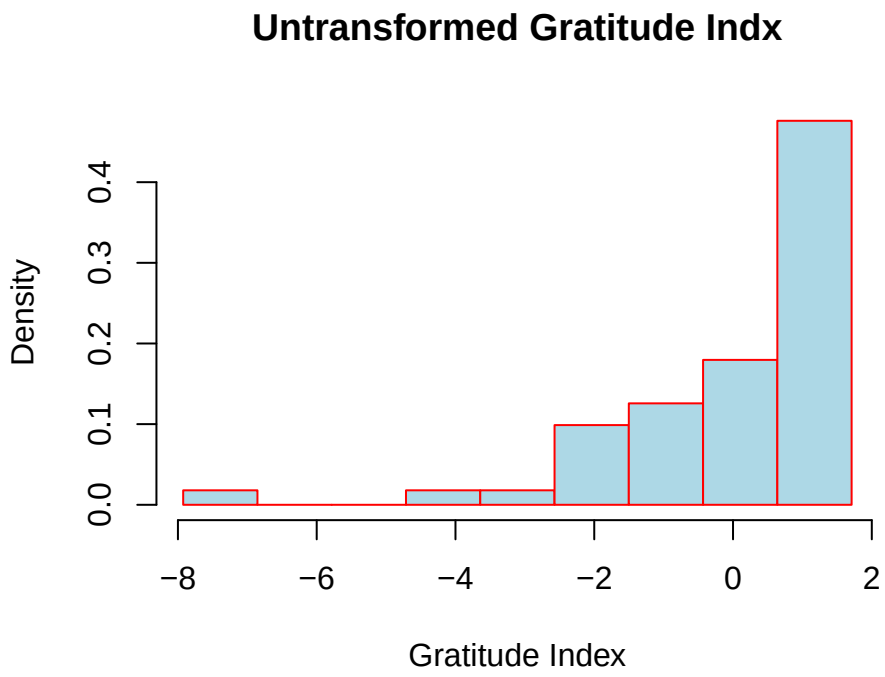


Figure 8: Histograms of the gratitude index scores. The distribution of the gratitude index is skewed to the left. We will perform a Box-Cox transformation to fix the distributional issue in the regression residuals.

```
M1=cor(cbind(gr.idx, gratitude.new, sc.idx, selfcomp.sub))
#corrplot(M, type = "upper", method = "ellipse", main="Pairwise Correlation Plot: Self-Compassion Scale
corrplot.mixed(M1, lower.col = "purple", upper = "ellipse", number.cex = .7, tl.cex = 0.7)
```

From the above plot, the first PCA of self-compassion is negatively associated with individual items in the instrument. This implies that the negative scores of the PCA are appropriate indexes for self-compassion. This index based on the negative PCA will be added to the final data set in the next section.

Figure 9: Pair-wise correlation plot between individual survey items and the two first PCA scores extract from the two instruments.