Self-compassion and Gratitude Scales and Their Association with Demographics

Cheng Peng

11/25/2020

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

```
survey = read.csv("w09-SurveyDataCsvFinal.csv", head = TRUE)
# names(survey)
```

The original survey data have three components, self-compassion scale and 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 impute 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 as 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 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 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 maintain the sample size and the statistical power of all subsequent association analysis. About 15 records in the data sets do not have demographic information. Therefore these records were deleted in 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-code 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 PCA and EFA for Survey Instruments

We perform both principal component analysis (PCA) and the exploratory factor analysis (EFA). The number We next try several exploratory factor analysis models to find the best model that explains the most variation of the data.

Different methods for finding the number of components/factors to retain in an exploratory principal component or factor analysis are used in this analysis. The classical ones are the Kaiser rule, the parallel analysis, and the usual scree test. Non-graphical solutions by Raiche et al. (2013) to the Cattell subjective scree test are also proposed: an acceleration factor (AF) and the index of the optimal coordinate OC. The acceleration factor indicates the turning point of the scree plot. It corresponds to the acceleration of the curve, i.e. the second derivative. The optimal coordinates are the extrapolated coordinates of the previous eigenvalue that allow the observed eigenvalue to go beyond this extrapolation.

Our objective is to aggregate the information to a few principle components that carry as much information as possible, then use simple statistical analysis to characterize the behavior of the resulting PCA or EFA scores and relevant association analyses. Therefore, the standard orthogonal rotation is used to avoid the potential correlation between the resulting PCA or EFA scores.

Whether PCA or EFA will be used and how many of the PCAs or EFAs will be selected will be based on the well-received analytical methods and subjective judgment on the proportion of variables explained by the selected PCAs or EFAs. In case, both PCA and EFA give similar results in terms of the proportion of the total variation, we will go with PCA since PCA's estimation of loading does not assume distribution of normality to define the maximum likelihood objection. This will reduce the risk of model misspecification.

3.1 Some R Functions For Extracting Information from PCA and EFA

We defined and modified some R functions to extract specific information from the PCA or EFA analyses to address the analytic questions outlined in the previous section.

My.plotnScree produce a plot that summarized the results of four commonly used methods used for identifying the number of components or factors to be retained for exploratory analyses. This graphics function was modified from the *plotnScree()* from library **nFactors**.

```
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
```

```r
    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 produce 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")
```

3.2 Self-compassion Index

We start with some correlation plot 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)
###
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
```

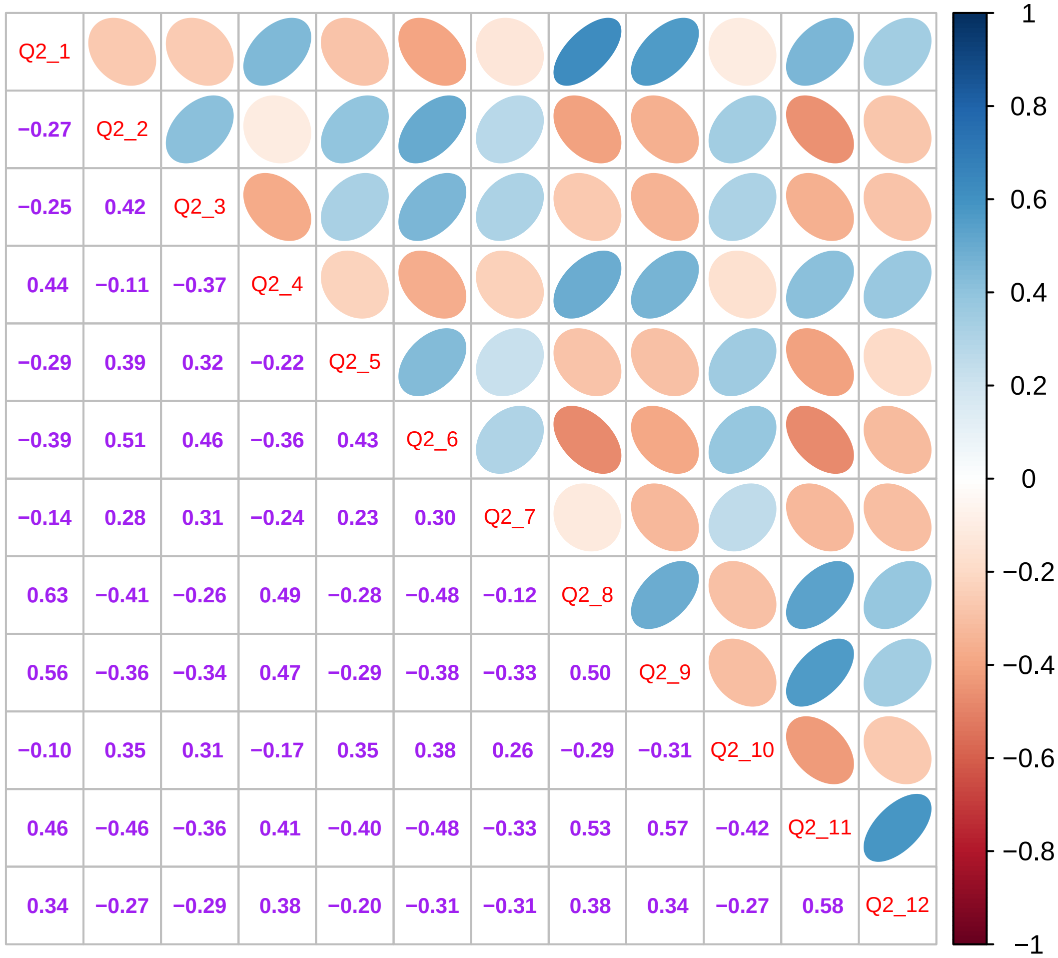


Figure 1: The pairwise correlation plot reveals the potential relevance of PCA. The shape of an ellipse represents the correlation. The skinnier the ellipse, the higher the correlation. The direction reflects whether a correlation is positive or negative. The off-diagonal direction implies s positive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

```
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)
```

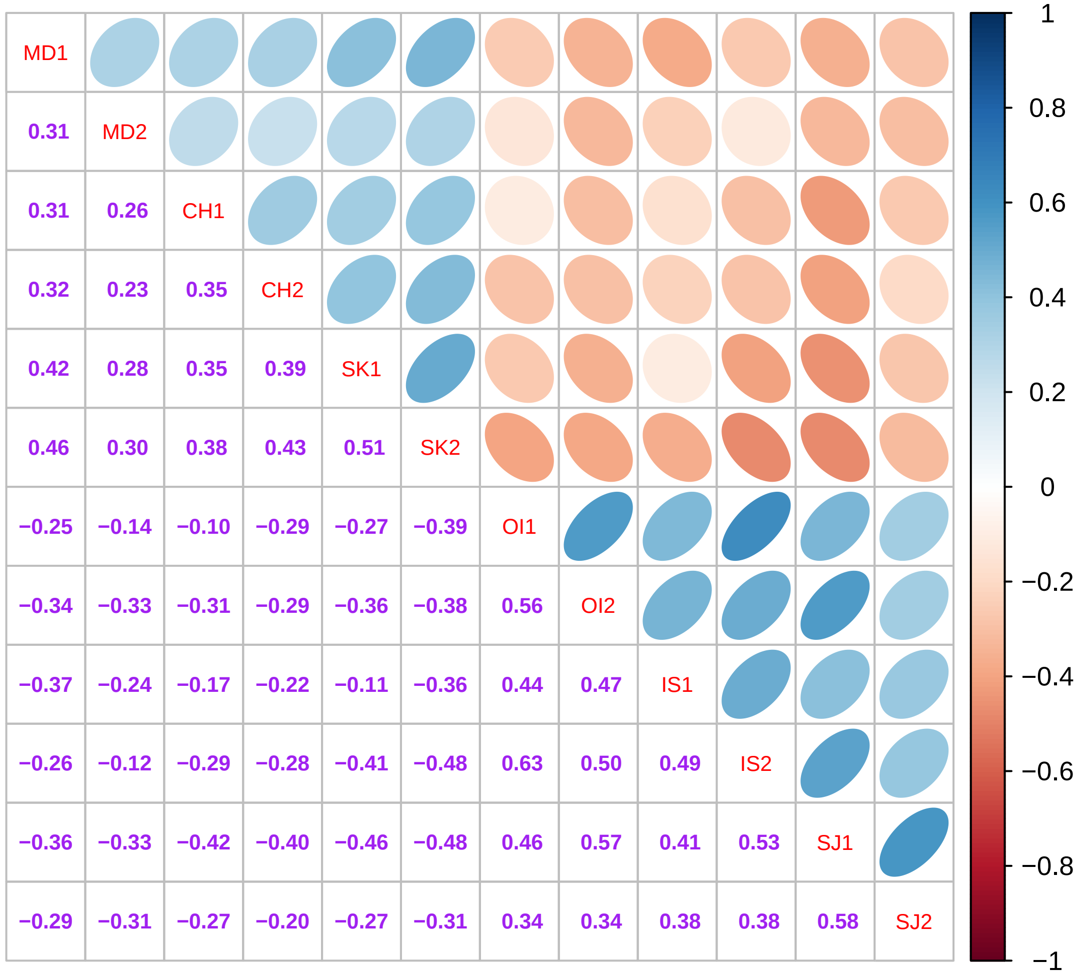


Figure 2: The pairwise correlation plot reveals the potential relevance of PCA. Group items based on self-compassion and self-coldness

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)
```

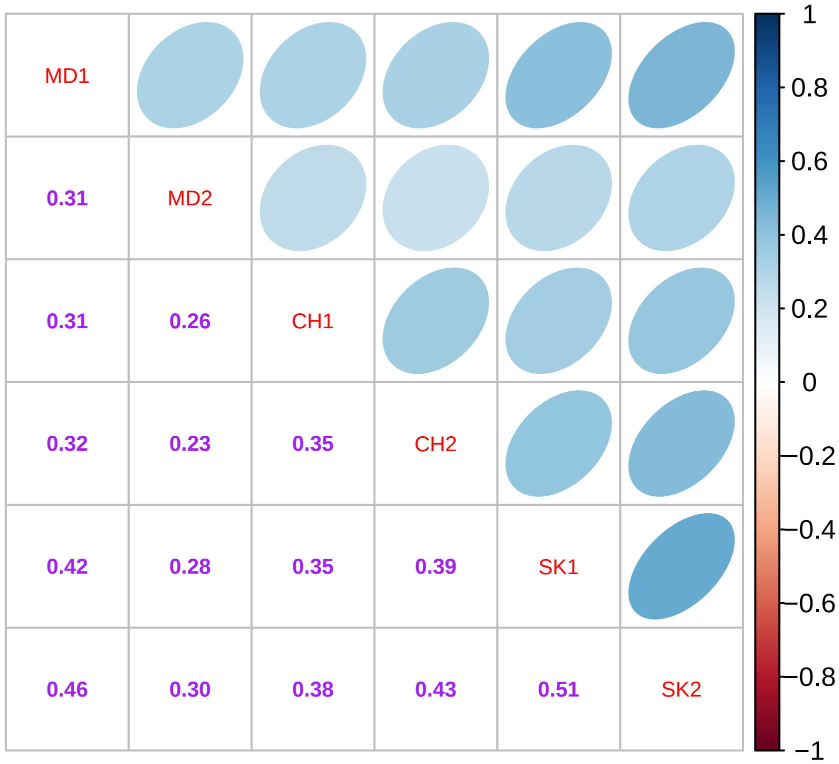


Figure 3: Pairwise correlation based on negatively adjusted subscales. The shape of an ellipse represents the correlation. The skinnier the ellipse, the higher the correlation. The direction reflects whether a correlation is positive or negative. The off-diagonal direction implies s positive correlation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

3.2.1 Intern Consistency

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

Next we find the Cronbach alpha and it 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 the Cronbach's alpha is 0.77 with 95% confidence interval (0.70, 0.84) suggesting that the items in the self-compassion instrument have relatively high internal consistency.

3.2.2 PCA Extraction and Number of PCA Determination

The number of PCAs selected for the 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.

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

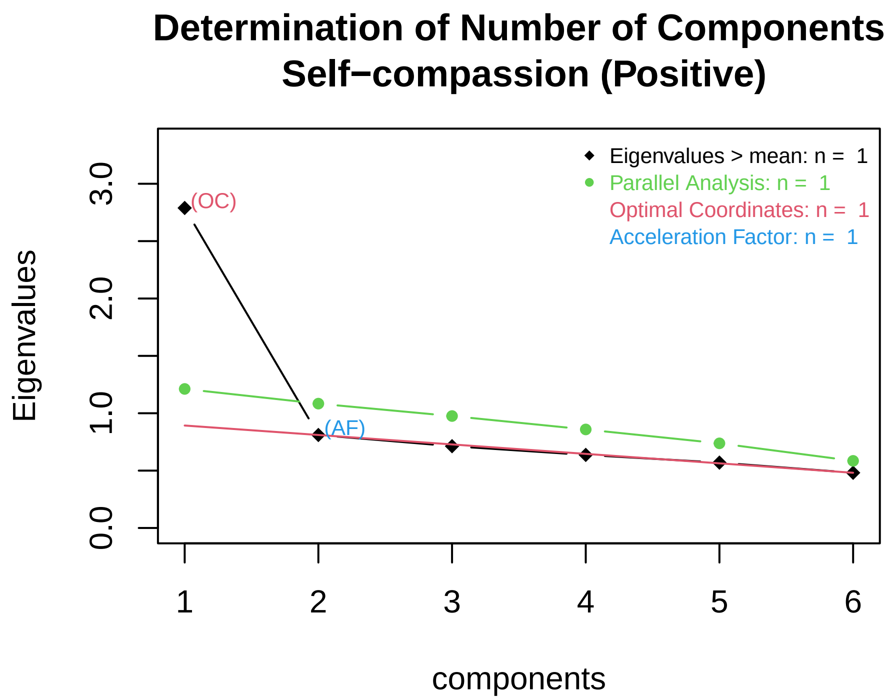


Figure 4: Different methods of identification of the number of principal components to be retained in exploratory analysis: Kaiser's eigenvalue rule, Raiche et al Monte Carlo simulation method (parallel analysis), optimal coordinate (OC) index, and accelerate factor (AF) method.

Figure 2 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.

```r
#
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 2: 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

	PC1	PC2
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
    principal component in the self-compassion survey.")
```

Table 3: Cumulative and proportion of variances explained by each 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 using PCA method and extract the first two principle components for the future analysis. Table 1 shows the factor loadings of the first two principle components. We can see that each of the original items contribute 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 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
)
```

3.3 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.

```
##
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)
```

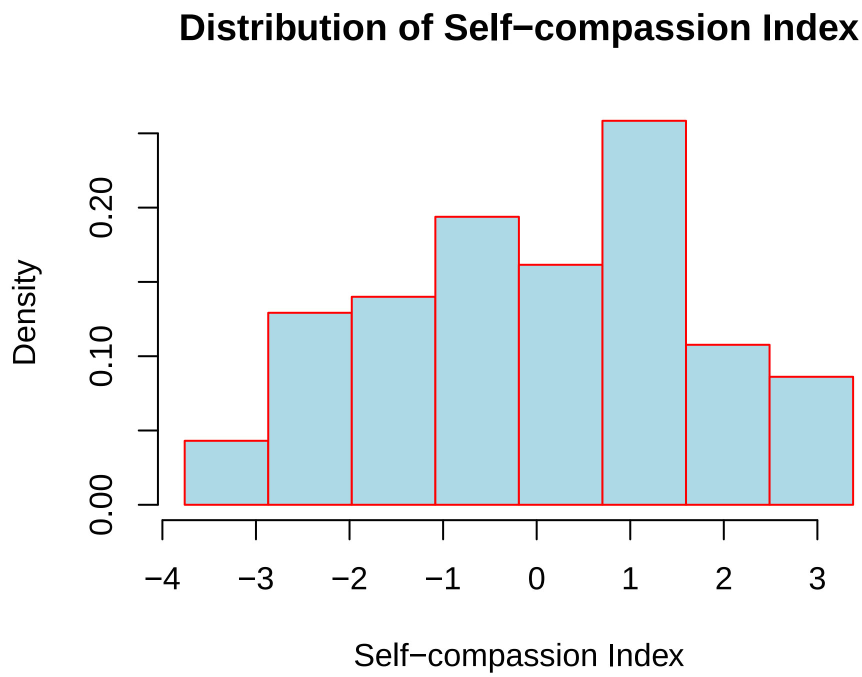


Figure 5: Histogram of the first principle component extract from the self-compassion survey.

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.

3.3.1 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 4: Confodence Interval of Cranbach Alpha

LCI	alpha	UCI
0.7280232	0.795529	0.8508849

We can see that the 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.

Figure 6: 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.

3.3.2 PCA Extraction and Number of PAC Determination

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

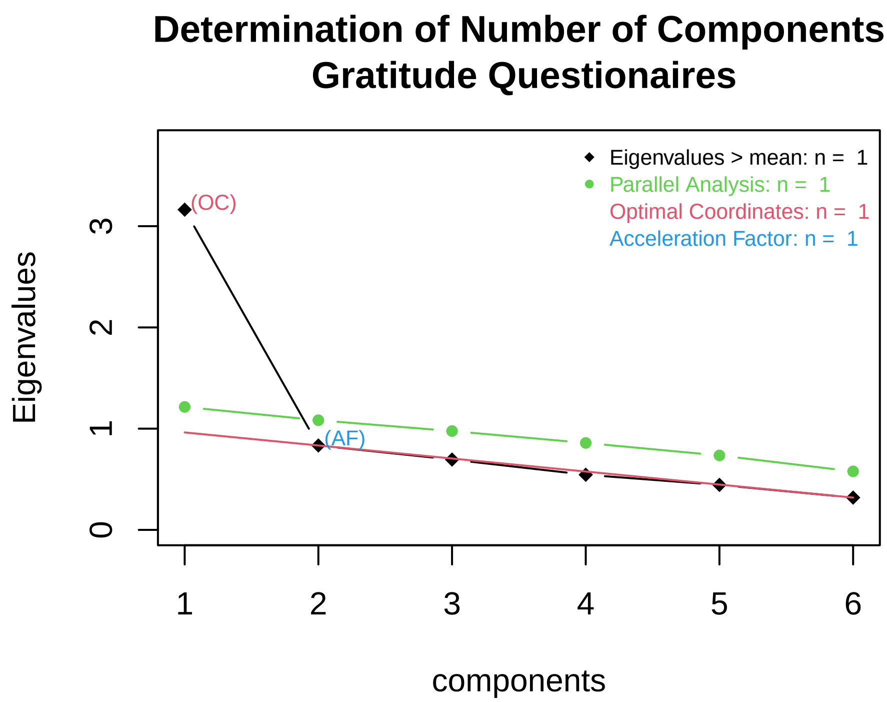


Figure 7: Different methods of identification of number of principle components to be retained in exploratory analysis for the gratitude survey instrument: Kaiser's eigenvalue rule, Raiche et al Monte Carlo simulation method (parallel analysis), optimal coordinate (OC) index, and accelerate factor (AF)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

Figure 5 indicates retaining one PCA is sufficient for future exploratory analyses. As we did in the self-compassion survey instrument, we extract 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
    proportion of variation explained by the corresponding PCAs in the
    Gratitude Questionaire Survey.")
```

Table 5: Factor loadings of the first few PCAs and the cumulative 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

	PC1	PC2
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 factors 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 a orthogonal rotation (transformation), we can adjust the direction of the coordinate system to make the PCA scores meaningful index for subsequent association analysis. The is the plot of the pairwised association between the individual items and the two new PCA scores.

```
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 that 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 the self-compassion. This index based on the negative PCA will be added to the final data set in the next section.

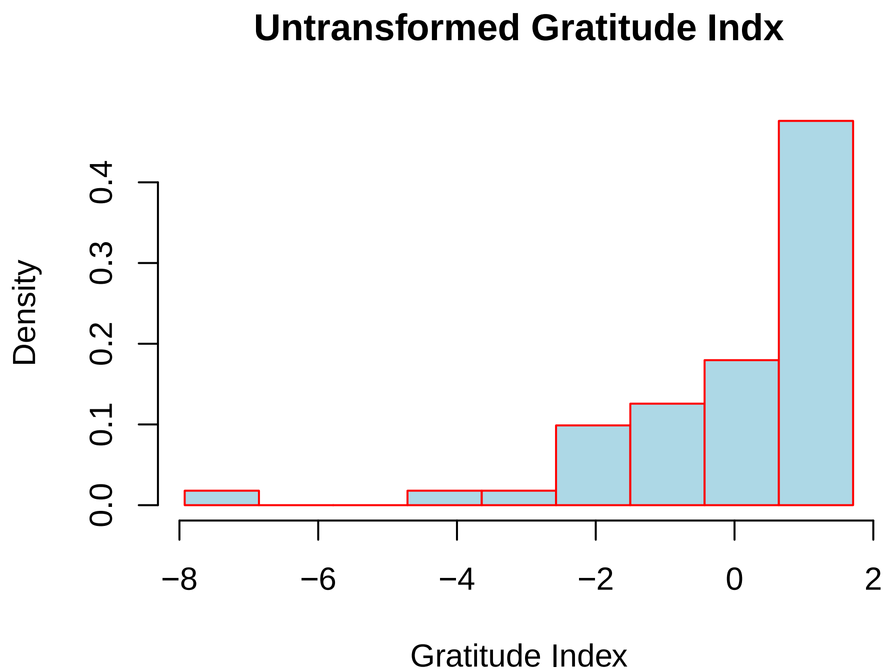


Figure 8: Histograms of the gratitude index scores. The distribution of the gratitude index is skewed to the left. We will perform a Box-Cox transformation to fix the distributional issue in the regression residuals.

3.4 Final Analytic Data Set

We now create a dataframe that contains PCAs based on self-compassion and gratitude indexes and the demographics.

```
final.analytic.data = new.demographics
final.analytic.data$sc.idx=sc.idx
final.analytic.data$gr.idx=gr.idx
final.analytic.data$lg.gr.idx=(gr.idx)^(1/3)
final.analytic.data$selfcomp.sum = selfcomp.sum
final.analytic.data$selfcomp.avg = selfcomp.avg
# final.analytic.data
## wrote the final analytic data frame to local drive as csv
write.csv(final.analytic.data,'w09-final-analytic-data.csv')
```

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

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

```
)
```

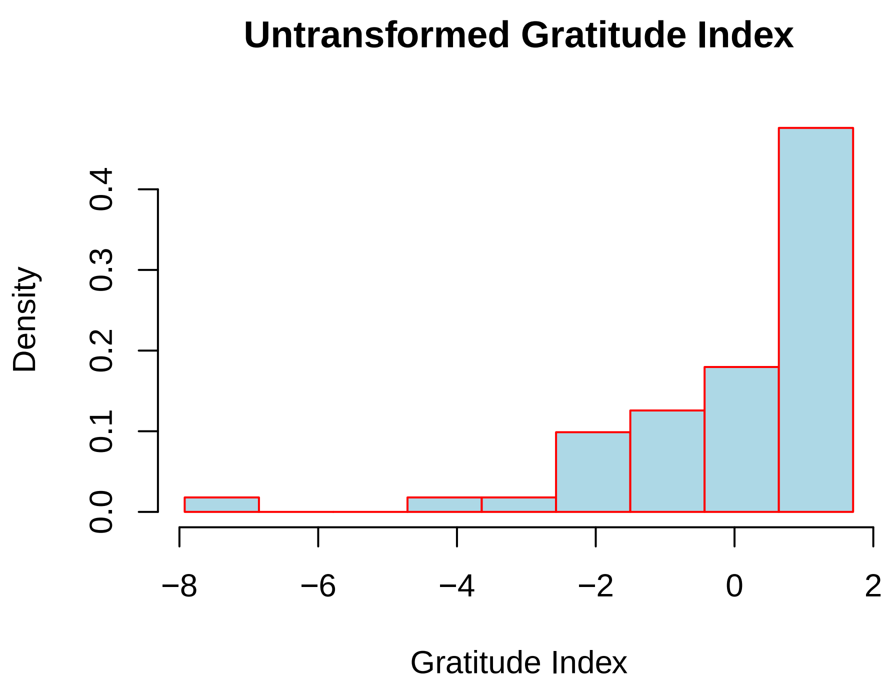


Figure 10: Histograms of the gratitude index scores. The distribution of the gratitude index is skewed to the left. We will perform a Box-Cox transformation to fix the distributional issue in the regression residuals.

In next section, we focus on building regression models to address the research questions.

4 Association Aanalyses

Three families of linear regression models will be built in this section. The final models in each of the three families will be presented to address the association between self-compassion and other variables of interests.

4.1 Association between Self-Compassion (raw Sum) and Demographics

The following R function reports the common goodness of fit measures to model selection.

```
gof.lm = function(mod.subj, in.p, ex.p, resp="Sum"){
    # mod.subj = name of the linear model
    # in.p = inclusion p value
    # ex.p = exclusion p vlue
    sum.obj = ols_step_forward_p(mod.subj, pent = in.p, prem = ex.p)
    summary.table=cbind(Predictor = sum.obj$predictors,
             Mallows_cp =round(sum.obj$mallows_cp,3),
             R.square = round(sum.obj$rsquare,3),
             SBIC = round(sum.obj$sbic,3),
             Adj.Rsq = round(sum.obj$adjr,3),
             RMSE = round(sum.obj$rmse,3),
             ACI = round(sum.obj$aic,3),
```

```
            SBC = round(sum.obj$sbc,3))
    ##
    kable(summary.table, caption=paste("Summary Table of Goodness of Fit Measures", resp))
}
```

The response variable will be the simple sum of a item scores.

```
final.data=read.csv("w09-final-analytic-data.csv", head = TRUE)
final.data.01 = final.data[,(-c(1,18, 19))]
null_model_sum = lm(selfcomp.sum ~ grp.age + home.size + poli.affil + spirituality +
    grp.edu + urbanity, data = final.data.01)
#ols_step_forward_p(null_model_sum, pent = .3, prem = 0.4)
gof.lm(null_model_sum, in.p = .3, ex.p= 0.4, resp=": Response = Sum")
```

Table 7: Summary Table of Goodness of Fit Measures : Response = Sum

Predictor	Mallows_cp	R.square	SBIC	Adj.Rsq	RMSE	ACI	SBC
spirituality	4.727	0.041	284.369	0.032	3.85	579.536	587.469
home.size	3.931	0.067	283.703	0.048	3.817	578.722	589.299
urbanity	2.451	0.099	282.414	0.062	3.789	579.109	594.975
grp.age	1.171	0.129	281.356	0.075	3.764	579.585	600.74

The results based on the sum of the raw item scores produce smaller R squares indicating that the the of raw item score is poorer than the PCA index. Next, we also try the mean item score and expect a similar results.

4.2 Association of Between Self-compassion (mean item scores) and Demographics

The response variable will be the mean item scores.

```
final.data=read.csv("w09-final-analytic-data.csv", head = TRUE)
final.data.01 = final.data[,(-c(1,18, 19))]
null_model_avg = lm(selfcomp.avg ~ grp.age + home.size + poli.affil + spirituality +
    grp.edu + urbanity, data = final.data.01)
gof.lm(null_model_avg, in.p = .3, ex.p= 0.4, resp=" : Response =  Mean")
```

Table 8: Summary Table of Goodness of Fit Measures : Response = Mean

Predictor	Mallows_cp	R.square	SBIC	Adj.Rsq	RMSE	ACI	SBC
spirituality	4.727	0.041	-88.317	0.032	0.642	206.85	214.783
home.size	3.931	0.067	-88.983	0.048	0.636	206.036	216.613
urbanity	2.451	0.099	-90.272	0.062	0.632	206.423	222.289
grp.age	1.171	0.129	-91.33	0.075	0.627	206.899	228.054

4.3 Association between Self-compassion and Demographics

In this section, we build and search the best regression model to detect the potential association between the self-compassion index and the demographics. we use an automatic variable selection procedure to find the final model that contains significant variables. The criterion used in the variable selection procedure is the p-value. Including all variable with p-value < 0.3 and excluding those variables with p-values > 0.4. The following gives the list of candiate variables for the final model.

```r
final.data=read.csv("w09-final-analytic-data.csv", head = TRUE)
final.data.01 = final.data[,(-c(1,18, 19))]
null_model = lm(sc.idx ~ grp.age + home.size + poli.affil + spirituality +
    grp.edu + urbanity, data = final.data.01)
#nmod.sum = ols_step_forward_p(null_model, pent = .3, prem = 0.4)
gof.lm(null_model_avg, in.p = .3, ex.p= 0.4, resp="Sum")
```

Table 9: Summary Table of Goodness of Fit Measures Sum

Predictor	Mallows_cp	R.square	SBIC	Adj.Rsq	RMSE	ACI	SBC
spirituality	4.727	0.041	-88.317	0.032	0.642	206.85	214.783
home.size	3.931	0.067	-88.983	0.048	0.636	206.036	216.613
urbanity	2.451	0.099	-90.272	0.062	0.632	206.423	222.289
grp.age	1.171	0.129	-91.33	0.075	0.627	206.899	228.054

Since the education level and political affiliation are of practical interest, we include these two into the final model.

```r
group.mean = function(resp.var, by.var){
  aggregate(resp.var,                 # Specify data column
          by = list(by.var),              # Specify group indicator
          FUN = mean)
}

group.sd = function(resp.var, by.var){
  aggregate(resp.var,               # Specify data column
          by = list(by.var),              # Specify group indicator
          FUN = sd)
}

group.len = function(resp.var, by.var){
  aggregate(resp.var,                # Specify data column
          by = list(by.var),              # Specify group indicator
          FUN = length)
}

### merge information to create descriptive statistics for EDA

merge3 = function(resp.var, by.var){
  anova.test = aov(resp.var ~ by.var, data = final.data.01)
  summary.dat =merge(merge(group.len(resp.var, by.var), group.mean(resp.var, by.var), by = "Group.1"),
                   group.sd(resp.var, by.var), by = "Group.1")
  anova.pval= c(round(summary(anova.test)[[1]][["Pr(>F)"]][1],4), rep(" ", (dim(summary.dat)[1])-1))
  out.table = cbind(summary.dat, anova.pval)
    names(out.table)=c("Category", "Size", "Mean", "Stdev", "p-value")
  out.table
}
```

The following tables provide the descriptive statistics about the self-compassion index by the categories of each candidate demographic variable.

```r
kable(merge3(resp.var =sc.idx, by.var = spirituality), caption = "Descriptive Statistics of Spiritualit
```

Table 10: Descriptive Statistics of Spirituality

Category	Size	Mean	Stdev	p-value
high	65	0.2604892	1.654386	0.0394
low	39	-0.4341487	1.624911	

```
kable(merge3(resp.var =sc.idx, by.var = home.size), caption = "Descriptive Statistics of Household Size
```

Table 11: Descriptive Statistics of Household Size

Category	Size	Mean	Stdev	p-value
1-2	50	-0.3479192	1.654103	0.0403
3+	54	0.3221474	1.634178	

```
kable(merge3(resp.var =sc.idx, by.var = urbanity), caption = "Descriptive Statistics of Urbanity")
```

Table 12: Descriptive Statistics of Urbanity

Category	Size	Mean	Stdev	p-value
rural	27	0.1817510	1.503999	0.3443
suburban	44	-0.2798687	1.770367	
urban	33	0.2244529	1.655277	

```
kable(merge3(resp.var =sc.idx, by.var = grp.age), caption = "Descriptive Statistics of Group Age")
```

Table 13: Descriptive Statistics of Group Age

Category	Size	Mean	Stdev	p-value
[1,23]	32	0.2058344	1.485451	0.0761
[24,30]	38	-0.4828086	1.711421	
[30,99]	34	0.3458831	1.707429	

```
kable(merge3(resp.var =sc.idx, by.var = grp.edu), caption = "Descriptive Statistics of Level")
```

Table 14: Descriptive Statistics of Level

Category	Size	Mean	Stdev	p-value
Adv.deg	12	-0.4479062	1.899614	0.5995
Assoc	26	-0.0109501	1.682387	
Bachelor	66	0.0857512	1.635714	

```
kable(merge3(resp.var =sc.idx, by.var = poli.affil), caption = "Descriptive Statistics of Political Aff
```

Table 15: Descriptive Statistics of Political Affiliations

Category	Size	Mean	Stdev	p-value
independent	31	0.3738856	1.391682	0.3748
ModDemocrats	31	-0.3654538	1.528507	
Republican	12	-0.1357411	2.054942	
StrongDemocrats	30	0.0455836	1.892458	

The next few tables provide the descriptive statistics about the gratitude index by the categories of each candidate demographic variable.

```
kable(merge3(resp.var =gr.idx, by.var = spirituality), caption = "Descriptive Statistics of Spirituality
```

Table 16: Descriptive Statistics of Spirituality

Category	Size	Mean	Stdev	p-value
high	65	0.3187487	1.699460	0.0176
low	39	-0.5312478	1.801781	

```
kable(merge3(resp.var =gr.idx, by.var = home.size), caption = "Descriptive Statistics of Household Size
```

Table 17: Descriptive Statistics of Household Size

Category	Size	Mean	Stdev	p-value
1-2	50	0.1559003	1.304555	0.3923
3+	54	-0.1443521	2.128359	

```
kable(merge3(resp.var =gr.idx, by.var = urbanity), caption = "Descriptive Statistics of Urbanity")
```

Table 18: Descriptive Statistics of Urbanity

Category	Size	Mean	Stdev	p-value
rural	27	0.2930024	1.681638	0.5666
suburban	44	-0.1735333	2.023479	
urban	33	-0.0083515	1.506073	

```
kable(merge3(resp.var =gr.idx, by.var = grp.age), caption = "Descriptive Statistics of Group Age")
```

Table 19: Descriptive Statistics of Group Age

Category	Size	Mean	Stdev	p-value
[1,23]	32	0.2762537	1.260125	0.5651
[24,30]	38	-0.1636728	1.918425	
[30,99]	34	-0.0770751	2.035715	

```
kable(merge3(resp.var =gr.idx, by.var = grp.edu), caption = "Descriptive Statistics of Level")
```

Table 20: Descriptive Statistics of Level

Category	Size	Mean	Stdev	p-value
Adv.deg	12	-0.7622257	2.519522	0.1225
Assoc	26	0.4849109	1.203123	
Bachelor	66	-0.0524390	1.782542	

```
kable(merge3(resp.var =gr.idx, by.var = poli.affil), caption = "Descriptive Statistics of Political Aff
```

Table 21: Descriptive Statistics of Political Affiliations

Category	Size	Mean	Stdev	p-value
independent	31	0.3084214	1.5047288	0.1984
ModDemocrats	31	-0.1844131	1.4970940	
Republican	12	0.6957153	0.9270148	
StrongDemocrats	30	-0.4064280	2.3956307	

The linear regression model will be based on the index scores created based on the principle component analysis.

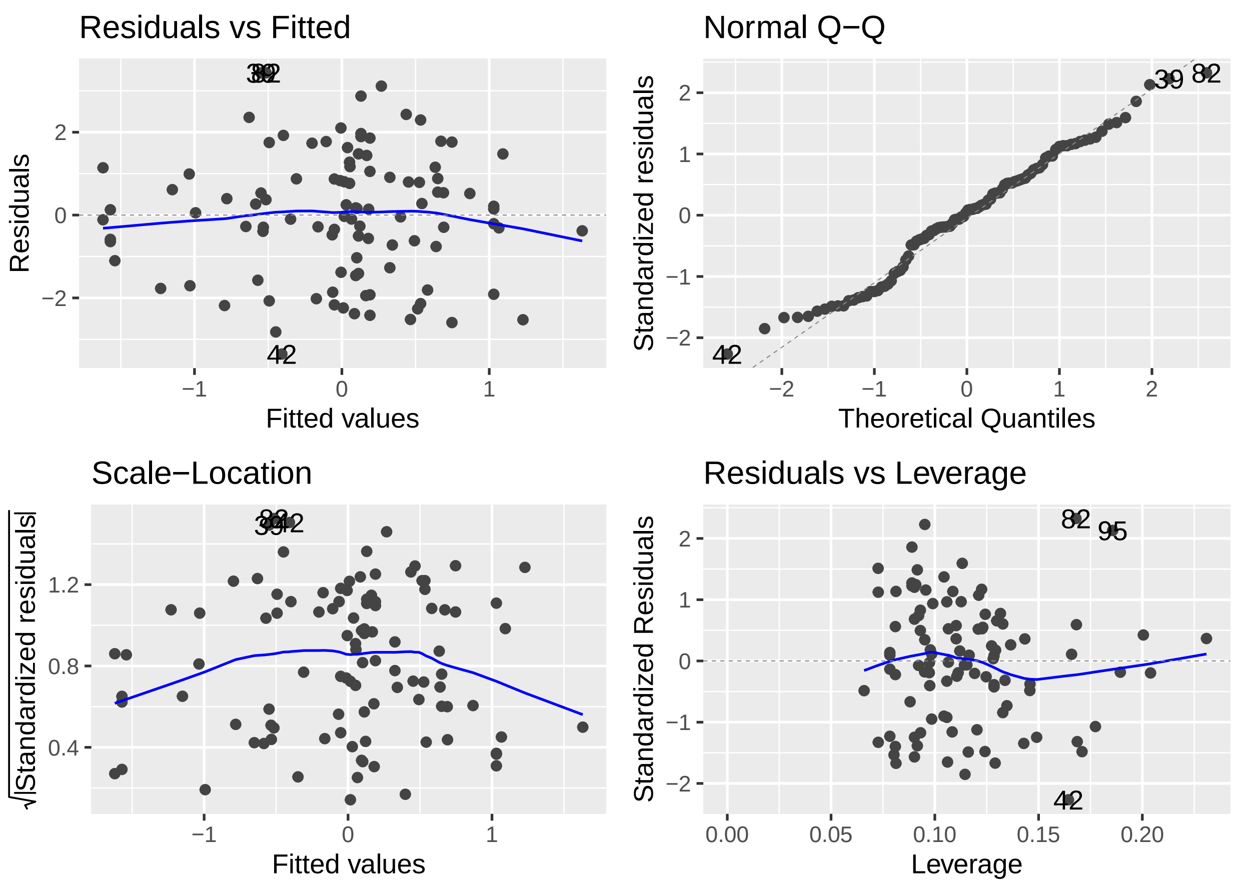
```
final.data=read.csv("w09-final-analytic-data.csv", head = TRUE)
final.data.01 = final.data[,(-c(1,18, 19))]
Final.model <- lm(sc.idx~ spirituality + home.size + urbanity + grp.age + poli.affil +
# full_model <- lm(sc.idx~., data = final.data.01)
# forward_model <- step(null_model,scope=list(lower=formula(null_model),
#                       upper=formula(full_model)), direction="both", trace = FALSE)
# full_model.adj <- lm(sc.idx~grp.age+grp.edu+home.size+poli.affil+spirituality, data = final.data.01)
##
kable(round(summary(Final.model)$coef,4), caption="Summary of Linear Regression model
     between self-compassion index and demographics")
```

Table 22: Summary of Linear Regression model between self-compassion index and demographics

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.7391	0.7956	0.9290	0.3553
spiritualitylow	-0.5783	0.3495	-1.6545	0.1014
home.size3+	0.4490	0.3482	1.2895	0.2005
urbanitysuburban	-0.3912	0.4106	-0.9527	0.3433
urbanityurban	0.6003	0.4717	1.2728	0.2063
grp.age[24,30]	-0.6946	0.4400	-1.5788	0.1178
grp.age[30,99]	-0.1076	0.4309	-0.2496	0.8035
poli.affilModDemocrats	-0.6458	0.4451	-1.4510	0.1502
poli.affilRepublican	-0.8125	0.5760	-1.4107	0.1617
poli.affilStrongDemocrats	-0.5095	0.4545	-1.1212	0.2651
grp.eduAssoc	-0.1056	0.6182	-0.1708	0.8647
grp.eduBachelor	-0.0500	0.5497	-0.0909	0.9278

Three variables were retained in the final model based on the step-wise model selection. The diagnostic plots of the above linear model are given in the following.

```
autoplot(Final.model)
```

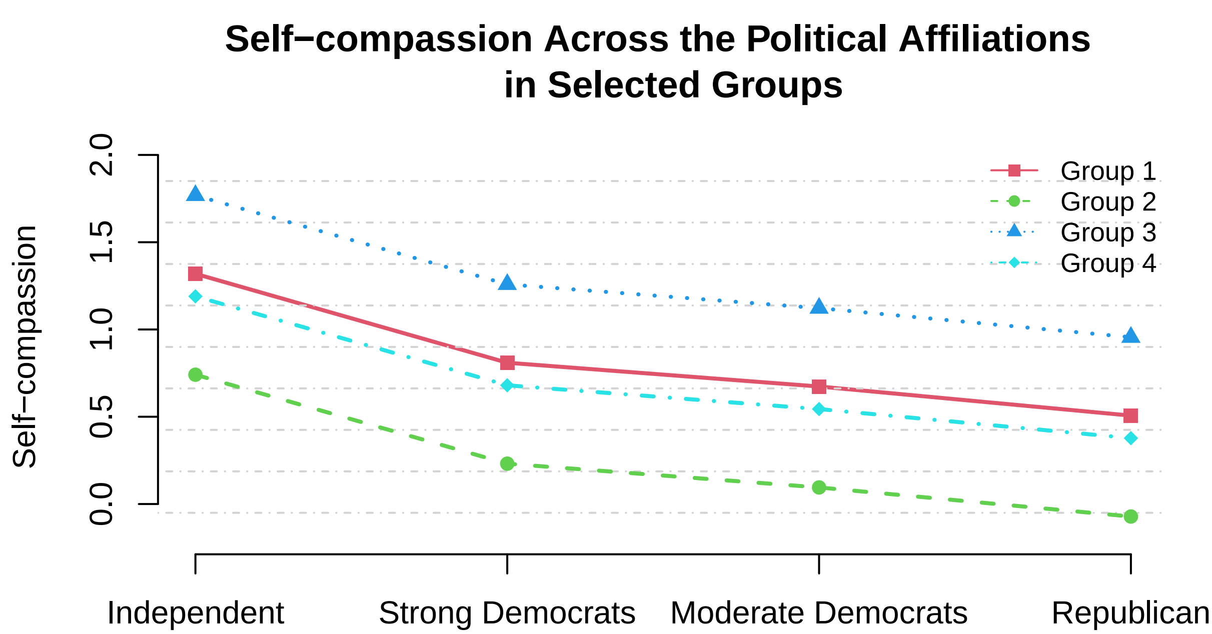


The diagnostic plots do not show serious violations of the model assumptions. We will summary this model in the next section.

```
# g1 (intercept + urban + age(<24) +bachelor): spirituality = high, househol = <3
g1.0 = 0.7391+0.63003-0.05 + 0          # indep
g1.1 = 0.7391+0.63003-0.05 - 0.5095     # strong D
g1.2 = 0.7391+0.63003-0.05 - 0.6458     # Mod. D
g1.3 = 0.7391+0.63003-0.05 - 0.8125     # republican
g1 = c(g1.0, g1.1, g1.2, g1.3)
## g2 (intercept + urban + age(<24) +bachelor): spirituality = low, househol = <3
g2.0 = 0.7391+0.63003-0.05 + 0      - 0.5783    # indep
g2.1 = 0.7391+0.63003-0.05 - 0.5095 - 0.5783     # strong D
g2.2 = 0.7391+0.63003-0.05 - 0.6458 - 0.5783     # Mod. D
g2.3 = 0.7391+0.63003-0.05 - 0.8125 - 0.5783    # republican
g2 = c(g2.0, g2.1, g2.2, g2.3)
## g3 (intercept + urban + age(<24) +bachelor): spirituality = high, househol >3
g3.0 = 0.7391+0.63003-0.05 + 0      + 0.4490    # indep
g3.1 = 0.7391+0.63003-0.05 - 0.5095 + 0.4490     # strong D
g3.2 = 0.7391+0.63003-0.05 - 0.6458 + 0.4490     # Mod. D
g3.3 = 0.7391+0.63003-0.05 - 0.8125 + 0.4490    # republican
g3 = c(g3.0, g3.1, g3.2, g3.3)
## g3 (intercept + urban + age(<24) +bachelor): spirituality = low, househol >3
```

```r
g4.0 = 0.7391+0.63003-0.05 + 0      + 0.4490 - 0.5783   # indep
g4.1 = 0.7391+0.63003-0.05 - 0.5095 + 0.4490 - 0.5783    # strong D
g4.2 = 0.7391+0.63003-0.05 - 0.6458 + 0.4490 - 0.5783    # Mod. D
g4.3 = 0.7391+0.63003-0.05 - 0.8125 + 0.4490 - 0.5783   # republican
g4 = c(g4.0, g4.1, g4.2, g4.3)
### line plot
# range(c(g1, g2, g3, g4))

plot(1:4, g1, ylim = c(-0.2, 2), axes = FALSE, bty="o", xlab = "",
     ylab ="Self-compassion", type="l", lty = 1, col = 2, lwd=2)
grid(NA, 10, lwd = 1, lty=4, col = "lightgray")   # grid only in y-direction
xlabel = c("Independent", "Strong Democrats", "Moderate Democrats", "Republican")
axis(1, at = 1:4, labels =xlabel)
axis(2)
par(bty="o")
points(1:4, g1, pch=15, col =2)
# g2
lines(1:4,  g2, lty = 2, col=3, lwd=2)
points(1:4, g2, pch =16, col=3)
# g3
lines(1:4,  g3, lty = 3, col=4, lwd=2)
points(1:4, g3, pch =17, col=4 )
# g4
lines(1:4,  g4, lty = 4, col=5, lwd=2)
points(1:4, g4, pch =18, col=5 )
#
legend("topright", c("Group 1", "Group 2", "Group 3", "Group 4"),
       lty=1:4, pch = 15:18, col = 2:5, bty = "n", cex = 0.8)
title("Self-compassion Across the Political Affiliations \n  in Selected Groups")
```



4.4 Association between Self-compassion and demographics adjusted by gratitude index

In this section, we build a model that is similar to the one in the previous subsection but it will be adjusted by the gratitude index.

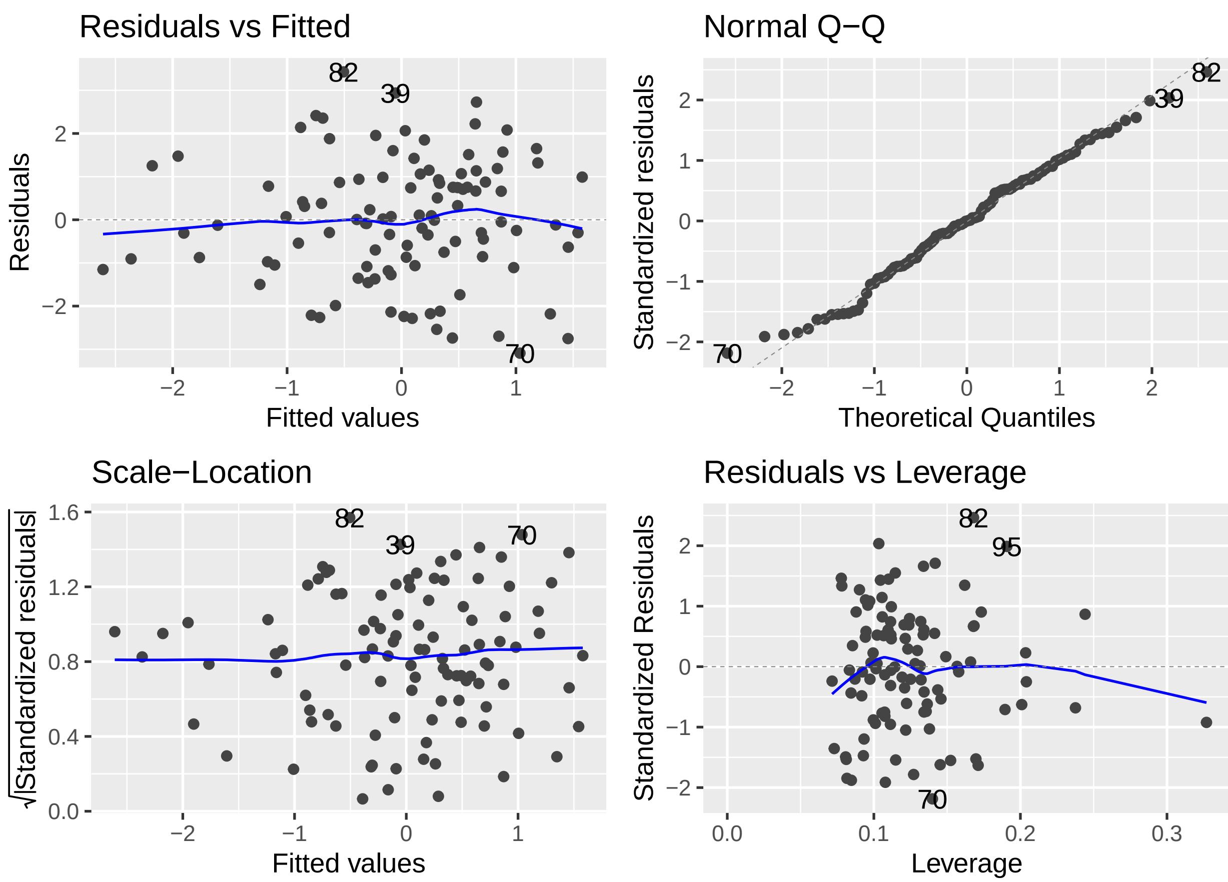
```
final.data.02 = final.data[,(-c(1, 18))]
# null_model.adj <- lm(sc.idx~1., data = final.data.02)
# full_model.adj <- lm(sc.idx~., data = final.data.02)
# forward_model.adj <- step(null_model.adj,scope=list(lower=formula(null_model.adj),
#                  upper=formula(full_model.adj)), direction="forward", trace = FALSE)
full_model.adj <- lm(sc.idx~spirituality + home.size + urbanity + grp.age + poli.affil +
##
kable(round(summary(full_model.adj)$coef,4), caption="Summary of Linear Regression model
      between self-compassion index and demographics adjusted by gratitude index")
```

Table 23: Summary of Linear Regression model between self-compassion index and demographics adjusted by gratitude index

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.6181	0.7496	0.8246	0.4118
spiritualitylow	-0.2470	0.3418	-0.7228	0.4717
home.size3+	0.6504	0.3325	1.9559	0.0535
urbanitysuburban	-0.3555	0.3867	-0.9193	0.3603
urbanityurban	0.5135	0.4446	1.1548	0.2512
grp.age[24,30]	-0.6156	0.4147	-1.4843	0.1412
grp.age[30,99]	-0.0044	0.4066	-0.0108	0.9914
poli.affilModDemocrats	-0.4811	0.4214	-1.1415	0.2567
poli.affilRepublican	-0.9626	0.5438	-1.7702	0.0800
poli.affilStrongDemocrats	-0.2643	0.4332	-0.6102	0.5433
grp.eduAssoc	-0.5345	0.5941	-0.8997	0.3706
grp.eduBachelor	-0.2915	0.5218	-0.5587	0.5778
gr.idx	0.3334	0.0931	3.5825	0.0005

The residual diagnostics plots are given in the following.

```
autoplot(full_model.adj)
```



Similar to what we saw in the diagnostic plots of the previous model, there is no significant violation of the assumption of the model assumption. This adjusted model will be used to address the association between the self-compassion index and the demographics.

4.4.1 Adjusted Relationship Between Self-compassion and Gratitude

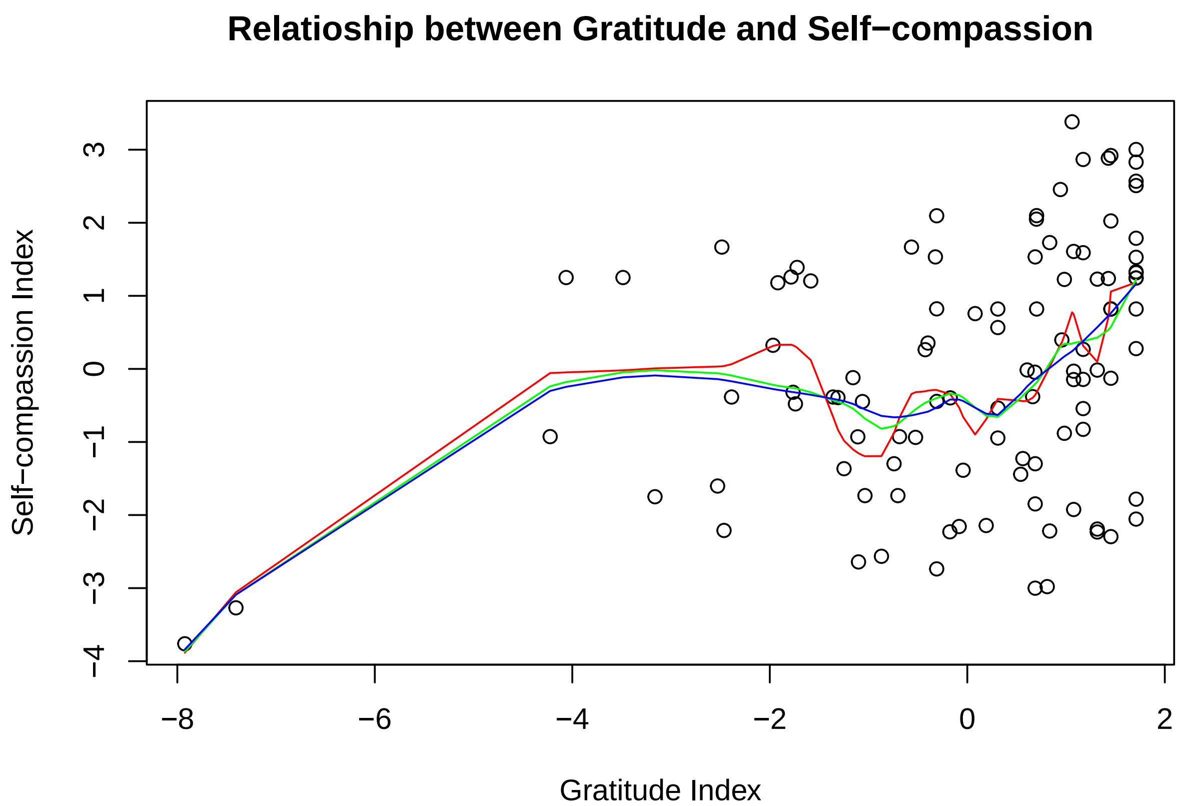
We create a plot showing the relationship between self-compassion and gratitude adjusted by demographics.

4.5 Association between Self-compassion Gratitude

We first plot the two index variables by a scatter plot and a non-parametric technique to fit the scatter plot to see the relationship between self-compassion and gratitude. We choose three different smoothing coefficients in the LOESS curve (LOcally WEighted Scatter-plot Smoother) plot which is given in the following.

```
final.loess = final.data[order(final.data$gr.idx),]
loessMod10 <- loess(sc.idx~gr.idx, data = final.loess, span=0.30) # 10% smoothing span
loessMod25 <- loess(sc.idx~gr.idx, data = final.loess, span=0.45) # 25% smoothing span
loessMod50 <- loess(sc.idx~gr.idx, data = final.loess, span=0.60) # 50% smoothing span
## prediction based on loess models with different bin widths
smoothed10 <- predict(loessMod10)
smoothed25 <- predict(loessMod25)
smoothed50 <- predict(loessMod50)
## base plot
plot(final.loess$gr.idx, final.loess$sc.idx, pch = 21,
      xlab="Gratitude Index", ylab = "Self-compassion Index",
     main = "Relatioship between Gratitude and Self-compassion")
```

```
lines(smoothed10, x=final.loess$gr.idx, col="red")
lines(smoothed25, x=final.loess$gr.idx, col="green")
lines(smoothed50, x=final.loess$gr.idx, col="blue")
```



This is a simple linear regression model.

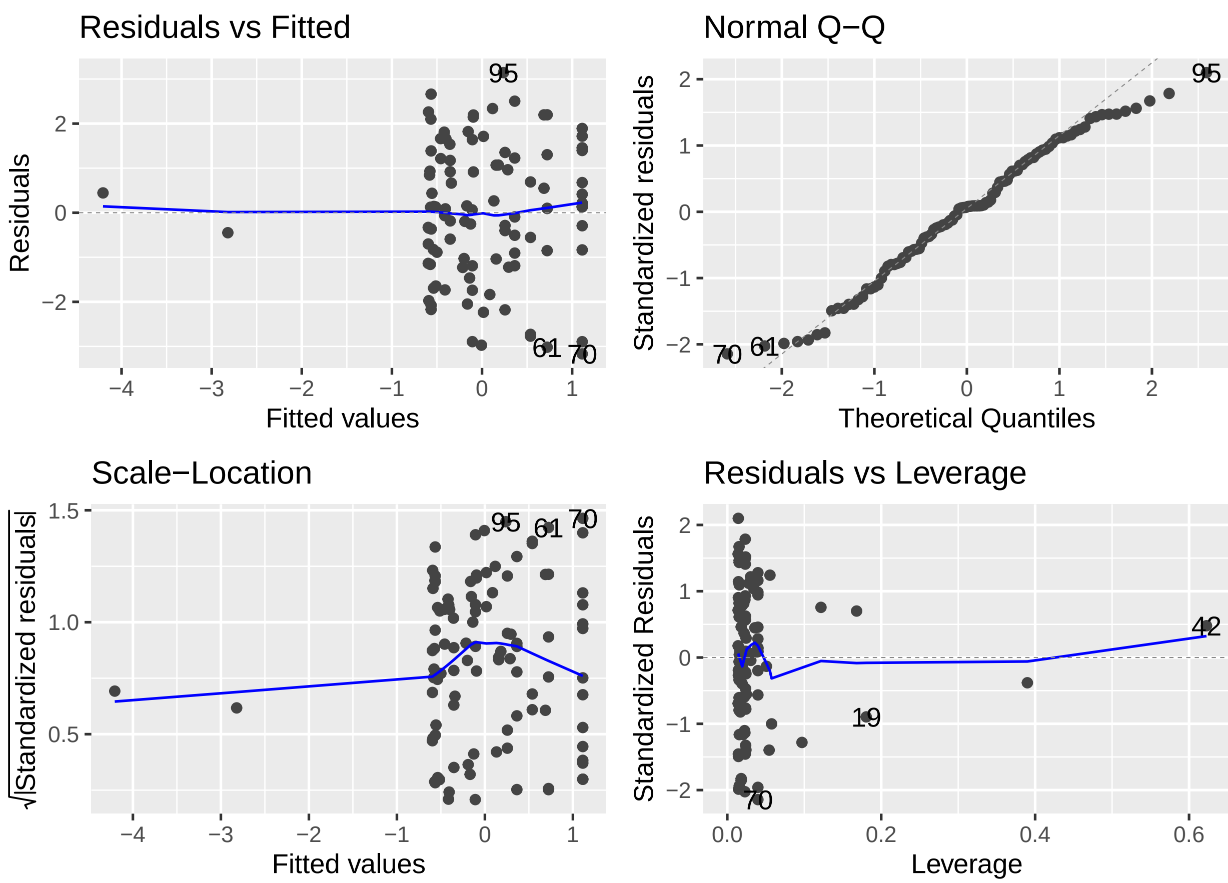
```
pca.reg = lm(sc.idx~poly(gr.idx, 3, raw = TRUE), data = final.data)
kable(round(summary(pca.reg)$coef,4), caption = "The marginal association between
                                    self-compassion and gratitude index")
```

Table 24: The marginal association between self-compassion and gratitude index

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.4861	0.2236	-2.1736	0.0321
poly(gr.idx, 3, raw = TRUE)1	0.3382	0.1210	2.7946	0.0062
poly(gr.idx, 3, raw = TRUE)2	0.2843	0.0901	3.1568	0.0021
poly(gr.idx, 3, raw = TRUE)3	0.0380	0.0116	3.2822	0.0014

The two variables are highly correlated. The diagnostic residual plots are given by

```
autoplot(pca.reg)
```



There are a few outliers but they are not influential. The models is appropriate.

5 References

Raiche, G., Walls, T. A., Magis, D., Riopel, M. and Blais, J.-G. (2013). Non-graphical solutions for Cattell's scree test. *Methodology*, 9(1), 23-29.

Golino, H. F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PloS One*, 12(6), e0174035.

Franklin, Scott B., Gibson, David J., Robertson, Philip A., Pohlmann, John T. and Fralish, James S. (1995). Parallel Analysis: a Method for Determining Significant Principal Components, *Journal of Vegetation Science*, Vol. 6(1): 99-106

Kotera, Y & Sheffield, D. (2020). Revisiting the Self-compassion Scale-Short Form: Stronger Associations with Self-inadequacy and Resilience, *SN Comprehensive Clinical Medicine*, 2: 761-769.

López, A, Sanderman, R., Ranchor, A. V. & Schroevers, M. J. (2018). Compassion for Others and Self-Compassion: Levels, Correlates,and Relationship with Psychological Well-being. *Mindfulness*, 9: 325 - 331.