COS-D419 Factor Analysis and Structural Equation Models 2023, Assignment 2

Päivimaria Mäkelä

2023-01-25

In my weekly report, the text of the original assignment is in *italics*, and my own additions are in normal font.

CFA & self-concept (SC)

Exercise 2.1

Specify and test the hypothesis given on the page 1 of the lecture material.

Draw conclusions based on the χ^2 statistic and the CFI, TLI, RMSEA, and SRMR indices.

What can you say about the parameter estimates?

Visualize the model.

Read in the data set:

Start by downloading the data file from Moodle to your Project folder!

```
library(tidyverse)
library(readr)

orig_data <- read_csv("ASC7INDM.CSV", show_col_types = FALSE)

# we will only use a subset of the data here:
SCdata <- orig_data %>% dplyr::select(starts_with("SDQ2N"))
#glimpse(SCdata)
str(SCdata)
```

```
## tibble [265 x 16] (S3: tbl_df/tbl/data.frame)
##  $ SDQ2N01: num [1:265] 6 6 4 5 6 5 1 2 5 4 ...
##  $ SDQ2N13: num [1:265] 5 6 6 5 5 5 6 1 5 6 ...
##  $ SDQ2N25: num [1:265] 4 6 6 5 5 5 1 6 6 3 ...
##  $ SDQ2N37: num [1:265] 6 6 2 6 4 3 6 4 6 6 ...
##  $ SDQ2N04: num [1:265] 3 6 6 5 3 3 4 4 6 6 ...
##  $ SDQ2N16: num [1:265] 4 6 4 6 4 2 6 4 6 5 ...
##  $ SDQ2N28: num [1:265] 4 6 6 5 4 4 6 4 6 6 ...
##  $ SDQ2N40: num [1:265] 6 6 3 6 4 4 6 6 6 6 ...
##  $ SDQ2N10: num [1:265] 2 5 6 5 4 4 1 6 5 4 ...
##  $ SDQ2N22: num [1:265] 6 6 5 6 6 4 6 6 6 6 ...
##  $ SDQ2N34: num [1:265] 1 6 4 3 5 5 1 1 5 4 ...
##  $ SDQ2N46: num [1:265] 5 6 5 5 6 6 6 5 6 6 ...
##  $ SDQ2N07: num [1:265] 6 6 6 6 3 4 5 3 6 5 ...
##  $ SDQ2N19: num [1:265] 6 6 6 6 4 5 6 4 6 6 ...
##  $ SDQ2N31: num [1:265] 6 6 3 6 4 4 6 4 6 6 ...
##  $ SDQ2N43: num [1:265] 6 6 1 5 5 4 5 6 6 6 ...
```

Explore the data (always do that!):

```r
library(psych)
library(dplyr)
library(knitr)
library(tidyr)
library(corrplot)
library(car)
library(psychTools)
library(kableExtra)
```

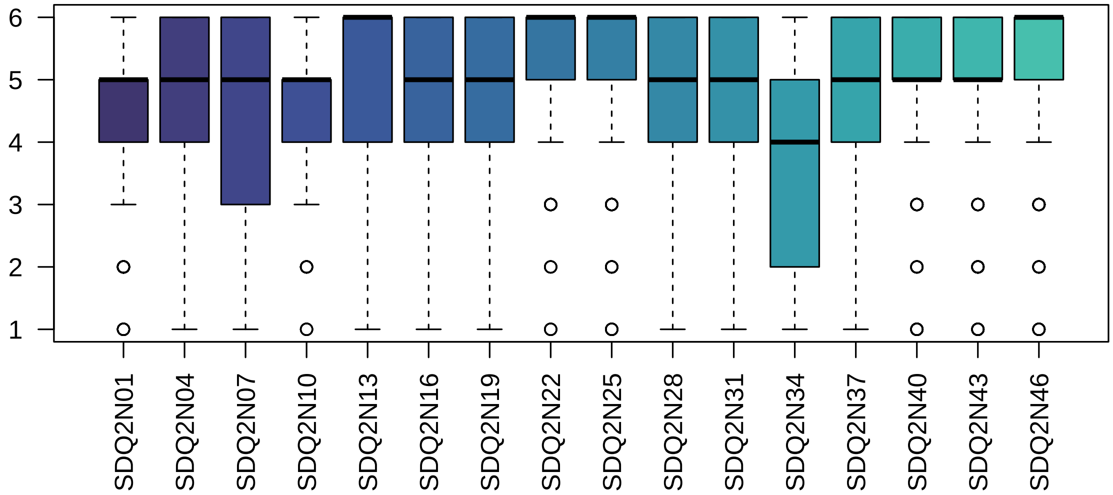
```r
# basic statistics:
#summary(SCdata)

SCdata_des  <- psych::describe(SCdata, ranges = FALSE, quant = c(0, 0.25, 0.5, 0.75, 1))
SCdata_des <- as.data.frame(SCdata_des)
names(SCdata_des) <- c("vars", "n", "Mean", "SD", "skew", "kurtosis", "se",
                   "Min.", "1st Qu.", "Median", "3rd Qu.", "Max.")

SCdata_des[, c("Median", "Mean", "SD", "Min.", "Max.")]  %>%
  round(digits=2) %>%
  kable(booktabs=T, align = "c", caption = "Some basic statistics") %>%
  kable_styling(full_width = F)
```

```r
#min(SCdata_des[, "Mean"])
#max(SCdata_des[, "Mean"])
#min(SCdata_des[, "SD"])
#max(SCdata_des[, "SD"])
```

```r
# boxplots
colors <- viridis::mako(32)
boxplot(SCdata[, order(names(SCdata))], main=" ", col=colors[9:24], ylim=c(1,6), yaxt="n", las=2)
axis(2, at=c(seq(1, 6, 1)), labels=c(seq(1, 6, 1)), las=2)
```

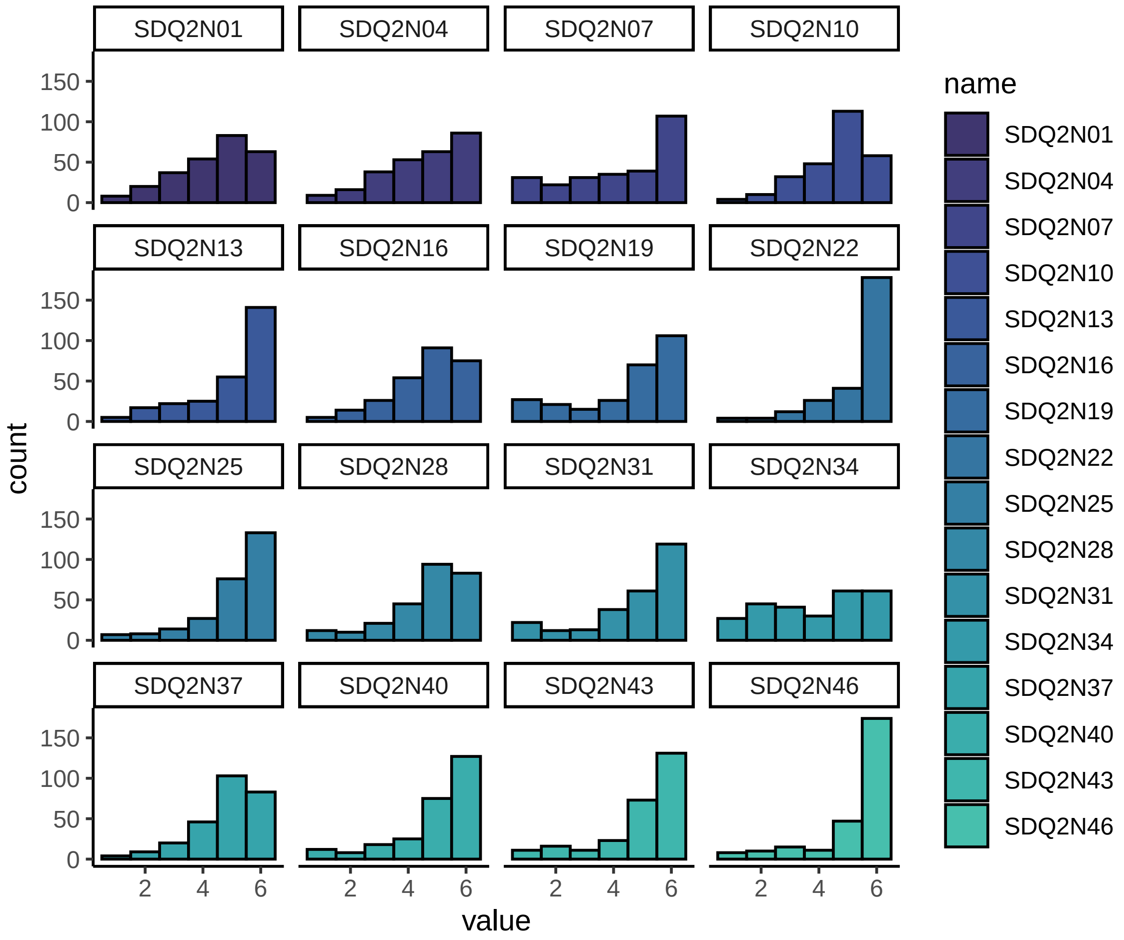
Table 1: Some basic statistics

	Median	Mean	SD	Min.	Max.
SDQ2N01	5	4.41	1.35	1	6
SDQ2N13	6	5.00	1.36	1	6
SDQ2N25	6	5.10	1.23	1	6
SDQ2N37	5	4.83	1.14	1	6
SDQ2N04	5	4.52	1.40	1	6
SDQ2N16	5	4.65	1.24	1	6
SDQ2N28	5	4.69	1.33	1	6
SDQ2N40	5	4.98	1.36	1	6
SDQ2N10	5	4.62	1.15	1	6
SDQ2N22	6	5.38	1.09	1	6
SDQ2N34	4	3.89	1.70	1	6
SDQ2N46	6	5.27	1.30	1	6
SDQ2N07	5	4.32	1.78	1	6
SDQ2N19	5	4.54	1.69	1	6
SDQ2N31	5	4.74	1.57	1	6
SDQ2N43	5	4.98	1.40	1	6

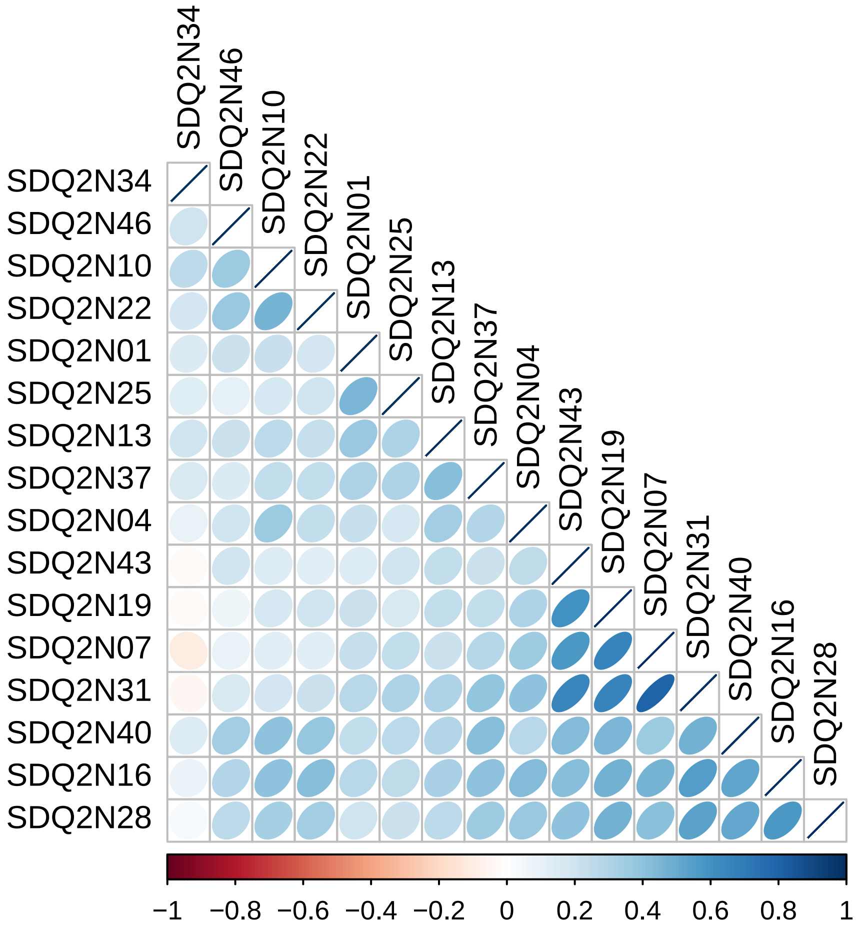


The mean values of the variables range from 3.89 to 5.38 and the standard deviations from 1.09 to 1.78. The distribution of almost all variables seems really skewed, and it's no wonder that most of the medians are 5 or 6 (the highest value is 6). Some basic statistics of the variables are shown in table 1.
To get a better idea of the distributions, I also want to see the histograms.

```
# histograms:
SCdata %>% pivot_longer(cols = everything()) %>%
  ggplot(aes(x = value, fill=name)) +
  geom_histogram(binwidth = 1, color="black") +
  scale_fill_manual(values=colors[9:24]) +
  theme_classic() +
  facet_wrap(~name, nrow = 4, ncol = 4)
```



```
# correlation plot:
SCdata %>%
  cor() %>%
  corrplot(method = 'ellipse', type = "lower", order = "hclust", tl.col = "black")
```



The correlation plot gives a good idea of the strength and direction of the correlations. However, sometimes it's good to see numbers.

It is a tricky business to create a neat and readable correlation matrix in R, especially when there are many variables. Table 2 shows my this week's solution to the problem. Please note that in the correlation plot the variables are arranged in clusters (similarity in correlations), but in the table they are in numerical order to improve readability. From the order of the variables in the correlationplot, we can almost conclude which variables load on the same factor. Only the variable SDQ2N04 seems to be in the wrong place.

```r
# correlation matrix:
SCdata[, order(names(SCdata))] %>%
  cor() %>%
  round(digits = 2) %>%
  kbl(booktabs=T, align = "c", caption = "Correlation matrix", format = "latex") %>%
  kable_styling(full_width = T, font_size = 8, latex_options = "scale_down") %>%
  landscape()
```

Table 2: Correlation matrix

	SDQ2N01	SDQ2N04	SDQ2N07	SDQ2N10	SDQ2N13	SDQ2N16	SDQ2N19	SDQ2N22	SDQ2N25	SDQ2N28	SDQ2N31	SDQ2N34	SDQ2N37	SDQ2N40	SDQ2N43	SDQ2N46
SDQ2N01	1.00	0.22	0.23	0.23	0.37	0.28	0.22	0.18	0.45	0.21	0.27	0.16	0.31	0.25	0.15	0.22
SDQ2N04	0.22	1.00	0.35	0.37	0.35	0.43	0.31	0.24	0.18	0.37	0.40	0.10	0.29	0.27	0.25	0.19
SDQ2N07	0.23	0.35	1.00	0.12	0.21	0.46	0.66	0.13	0.25	0.41	0.80	-0.11	0.29	0.37	0.58	0.09
SDQ2N10	0.23	0.37	0.12	1.00	0.26	0.41	0.18	0.47	0.18	0.34	0.19	0.26	0.25	0.41	0.14	0.36
SDQ2N13	0.37	0.35	0.21	0.26	1.00	0.33	0.25	0.23	0.30	0.26	0.31	0.20	0.42	0.29	0.25	0.21
SDQ2N16	0.28	0.43	0.46	0.41	0.33	1.00	0.48	0.43	0.26	0.57	0.55	0.09	0.41	0.52	0.43	0.29
SDQ2N19	0.22	0.31	0.66	0.18	0.25	0.48	1.00	0.20	0.17	0.47	0.66	-0.03	0.25	0.45	0.61	0.07
SDQ2N22	0.18	0.24	0.13	0.47	0.23	0.43	0.20	1.00	0.19	0.35	0.22	0.19	0.24	0.39	0.12	0.37
SDQ2N25	0.45	0.18	0.25	0.18	0.30	0.26	0.17	0.19	1.00	0.21	0.30	0.14	0.31	0.26	0.19	0.11
SDQ2N28	0.21	0.37	0.41	0.34	0.26	0.57	0.47	0.35	0.21	1.00	0.54	0.04	0.35	0.51	0.40	0.26
SDQ2N31	0.27	0.40	0.80	0.19	0.31	0.55	0.66	0.22	0.30	0.54	1.00	-0.05	0.40	0.48	0.65	0.16
SDQ2N34	0.16	0.10	-0.11	0.26	0.20	0.09	-0.03	0.19	0.14	0.04	-0.05	1.00	0.16	0.14	-0.03	0.21
SDQ2N37	0.31	0.29	0.29	0.25	0.42	0.41	0.25	0.24	0.31	0.35	0.40	0.16	1.00	0.43	0.21	0.15
SDQ2N40	0.25	0.27	0.37	0.41	0.29	0.52	0.45	0.39	0.26	0.51	0.48	0.14	0.43	1.00	0.44	0.34
SDQ2N43	0.15	0.25	0.58	0.14	0.25	0.43	0.61	0.12	0.19	0.40	0.65	-0.03	0.21	0.44	1.00	0.19
SDQ2N46	0.22	0.19	0.09	0.36	0.21	0.29	0.07	0.37	0.11	0.26	0.16	0.21	0.15	0.34	0.19	1.00

Define and estimate a CFA model:

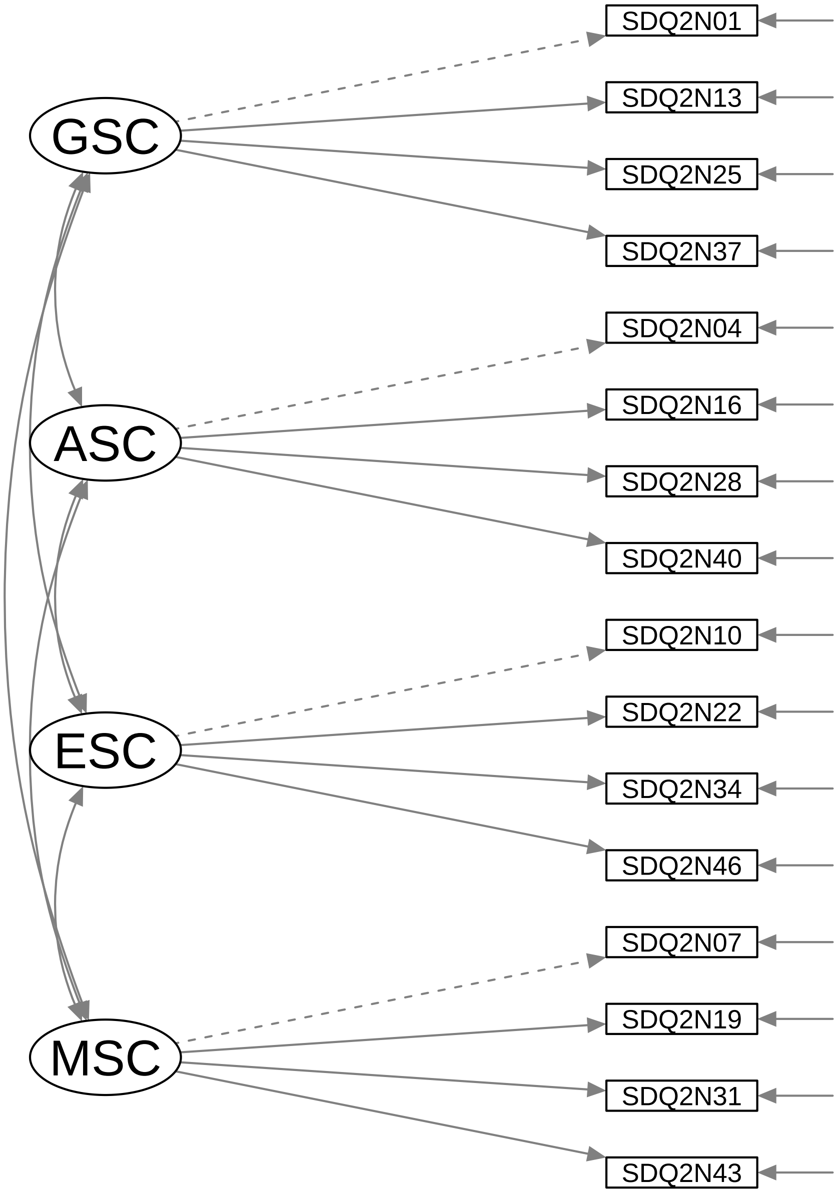
```
library(lavaan) # install.packages("lavaan")

# Define a CFA model using the lavaan package:
# NOTE: with the model definitions in lavaan syntax you have to
# SELECT all the code up to ' and then press Ctrl+Enter / Cmd+Enter
# when activating operations individually.

model1 <- '# CFA model of self-concept (SC):
           GSC =~ SDQ2N01 + SDQ2N13 + SDQ2N25 + SDQ2N37
           ASC =~ SDQ2N04 + SDQ2N16 + SDQ2N28 + SDQ2N40
           ESC =~ SDQ2N10 + SDQ2N22 + SDQ2N34 + SDQ2N46
           MSC =~ SDQ2N07 + SDQ2N19 + SDQ2N31 + SDQ2N43
          '

# Estimate the model using the data defined earlier:
cfa1 <- cfa(model1, data = SCdata)

# Numerical summary of the model:
#summary(cfa1, fit.measures = TRUE, standardized = TRUE)

#The summary is nice, but I don't want to print it.
```

```
#chi^2 statistic and the CFI, TLI, RMSEA, and SRMR indices.*
cfa1_res <- fitMeasures(cfa1, c("chisq", "df", "pvalue",
                                            "cfi", "tli", "rmsea",
                                            "rmsea.ci.lower", "rmsea.ci.upper",
                                            "rmsea.pvalue", "srmr"))
cfa1_res <- as.matrix(cfa1_res)
cfa1_res <- t(cfa1_res)
cfa1_res <- as.data.frame(cfa1_res)
```

Statistics and conclusions can be found in table 3 at the end of the weekly report.

Visualize the CFA model:

```
library(semPlot) # install.packages("semPlot")

# LISREL style (introduced by K. Jöreskog in the 1970s - still the standard):

# "some" more options added:
semPaths(cfa1, style = "lisrel", layout = "tree2", what = "path", whatLabels = "name",
    intercepts = FALSE, residuals = TRUE, thresholds = FALSE, reorder = FALSE,
    rotation = 2,
    latents = c("MSC","ESC", "ASC" ,"GSC"),
    sizeLat = 10, sizeLat2 = 5,
    manifests = rev(colnames(SCdata)),
    sizeMan = 10, sizeMan2 = 2
    )
```


SDQ2N01
SDQ2N13
SDQ2N25
SDQ2N37
GSC
SDQ2N04
SDQ2N16
SDQ2N28
SDQ2N40
ASC
SDQ2N10
SDQ2N22
SDQ2N34
SDQ2N46
ESC
SDQ2N07
SDQ2N19
SDQ2N31
SDQ2N43
MSC

Exercise 2.2

Specify and test these two additional hypotheses (again draw conclusions based on the χ^2 statistic and the CFI, TLI, RMSEA, and SRMR indices):

- *Hypothesis 2: SC is a two-factor structure consisting of GSC and ASC (so that the four GSC measures load onto the GSC and all other onto the ASC).*
- *Hypothesis 3: SC is a one-factor structure.*

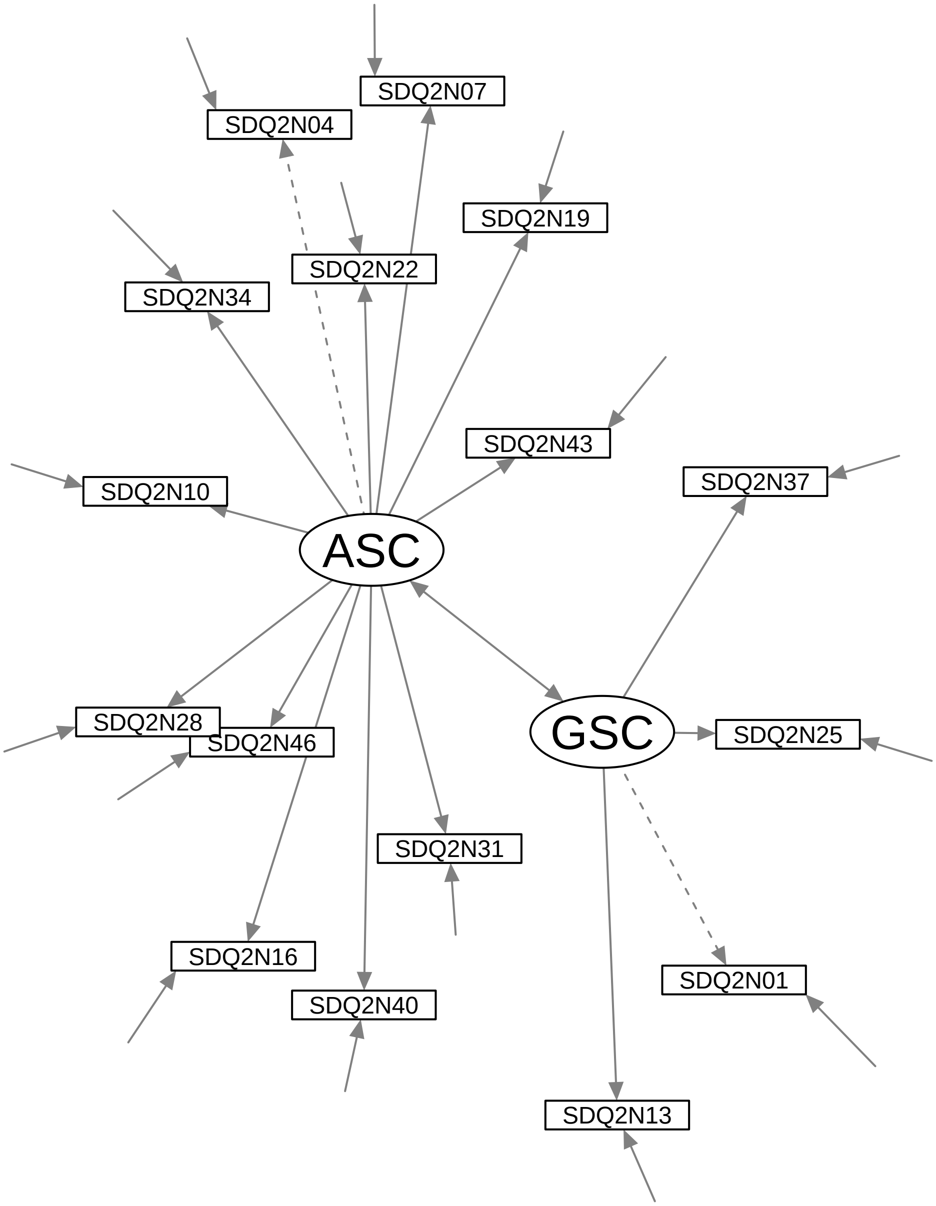
Visualize the models and compare them with the four-factor model analyzed in Exercise 2.1.

Hypothesis 2: SC is a two-factor structure consisting of GSC and ASC (so that the four GSC measures load onto the GSC and all other onto the ASC).**

```
model2 <- '# CFA model of self-concept (SC):
           GSC =~ SDQ2N01 + SDQ2N13 + SDQ2N25 + SDQ2N37
           ASC =~ SDQ2N04 + SDQ2N16 + SDQ2N28 + SDQ2N40
                  + SDQ2N10 + SDQ2N22 + SDQ2N34 + SDQ2N46
                  + SDQ2N07 + SDQ2N19 + SDQ2N31 + SDQ2N43
          '

# Estimate the model using the data defined earlier:
cfa2 <- cfa(model2, data = SCdata)

# Numerical summary of the model:
#summary(cfa2, fit.measures = TRUE, standardized = TRUE)
```

```
#chi^2 statistic and the CFI, TLI, RMSEA, and SRMR indices.*
cfa2_res <- fitMeasures(cfa2, c("chisq", "df", "pvalue",
                                            "cfi", "tli", "rmsea",
                                            "rmsea.ci.lower", "rmsea.ci.upper",
                                            "rmsea.pvalue", "srmr"))
cfa2_res <- as.matrix(cfa2_res)
cfa2_res <- t(cfa2_res)
cfa2_res <- as.data.frame(cfa2_res)
```

```
# "some" more options added:
semPaths(cfa2, style = "lisrel", layout = "spring", what = "path", whatLabels = "name",
    intercepts = FALSE, residuals = TRUE, thresholds = FALSE, reorder = FALSE,
    rotation = 2,
    latents = c("ASC" ,"GSC"),
    sizeLat = 10, sizeLat2 = 5,
    manifests = rev(colnames(SCdata)),
    sizeMan = 10, sizeMan2 = 2
    )
```

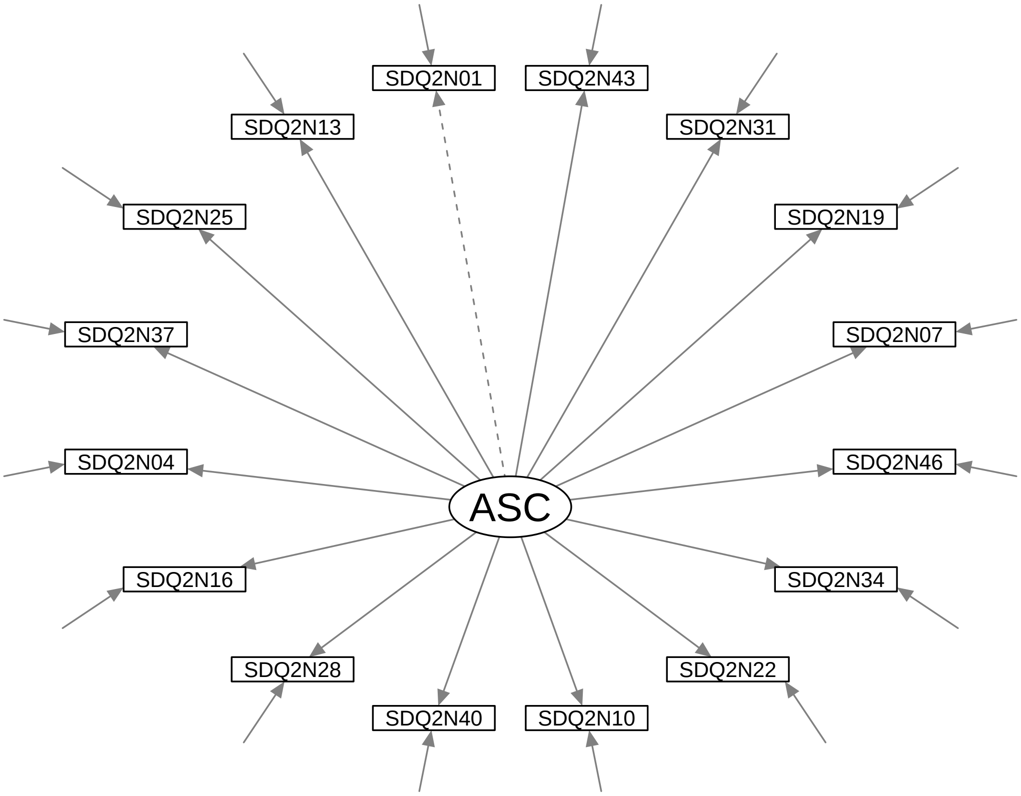

SDQ2N07
SDQ2N04
SDQ2N19
SDQ2N22
SDQ2N34
SDQ2N43
SDQ2N37
SDQ2N10
ASC
SDQ2N28
SDQ2N46
GSC
SDQ2N25
SDQ2N31
SDQ2N16
SDQ2N01
SDQ2N40
SDQ2N13

Hypothesis 3: SC is a one-factor structure.**

```
model3 <- '# CFA model of self-concept (SC):
           ASC =~ SDQ2N01 + SDQ2N13 + SDQ2N25 + SDQ2N37
                  + SDQ2N04 + SDQ2N16 + SDQ2N28 + SDQ2N40
                  + SDQ2N10 + SDQ2N22 + SDQ2N34 + SDQ2N46
                  + SDQ2N07 + SDQ2N19 + SDQ2N31 + SDQ2N43
          '

# Estimate the model using the data defined earlier:
cfa3 <- cfa(model3, data = SCdata)

# Numerical summary of the model:
#summary(cfa3, fit.measures = TRUE, standardized = TRUE)
```

```
#chi^2 statistic and the CFI, TLI, RMSEA, and SRMR indices.*
cfa3_res <- fitMeasures(cfa3, c("chisq", "df", "pvalue",
                                            "cfi", "tli", "rmsea",
                                            "rmsea.ci.lower", "rmsea.ci.upper",
                                            "rmsea.pvalue", "srmr"))
cfa3_res <- as.matrix(cfa3_res)
cfa3_res <- t(cfa3_res)
cfa3_res <- as.data.frame(cfa3_res)
```

```
# "some" more options added:
semPaths(cfa3, style = "lisrel", layout = "circle", what = "path", whatLabels = "name",
    intercepts = FALSE, residuals = TRUE, thresholds = FALSE, reorder = FALSE,
    rotation = 1,
    latents = c("ASC"),
    sizeLat = 10, sizeLat2 = 5,
    manifests = rev(colnames(SCdata)),
    sizeMan = 10, sizeMan2 = 2
    )
```



Draw conclusions based on the χ^2 statistic and the CFI, TLI, RMSEA, and SRMR indices.

What can you say about the parameter estimates?

```
# Combine statistics

cfa_res <- rbind(cfa1_res, cfa2_res, cfa3_res)

row.names(cfa_res) <- c("Model 1", "Model 2", "Model 3")

colnames(cfa_res) <- c("Test Statistic", "Degrees of Freedom", "P-Value",
                       "CFI", "TLI", "Estimate",
                       "90 Percent Lower C.I.", "90 Percent Upper C.I.",
                       "P-Value", "SRMR")

cfa_res %>%
```

Table 3: Factormodel statistics

	Chi-square Test			CFI/TLI		RMSEA				
	Test Statistic	Degrees of Freedom	P-Value	CFI	TLI	Estimate	90 Percent Lower C.I.	90 Percent Upper C.I.	P-Value	SRMR
Model 1	159.112	98	0	0.961	0.953	0.049	0.034	0.062	0.556	0.048
Model 2	457.653	103	0	0.776	0.739	0.114	0.103	0.125	0.000	0.101
Model 3	531.918	104	0	0.730	0.688	0.125	0.114	0.135	0.000	0.104

```
  round(digits=3) %>%
  kable(booktabs=T, align = "c", caption = "Factormodel statistics") %>%
  kable_styling(full_width = T) %>%
  add_header_above(c(" " = 1, "Chi-square Test" = 3, "CFI/TLI" = 2, "RMSEA" = 4, " " = 1))
```

Chi-Square Test of Model Fit of all three models indicates that H0 should be rejected (the fit of the data to the model is not adequate) However, this does not tell the whole truth, and if the other indices suggest that a model is ok, this does not need to be given much importance.

Comparative Fit Index (CFI) compares the hypothesized (H) and the baseline model (B). The lecture slides advised that well-fitting models have CFI over 0.95. The three models tested this week, only model 1 (four factor model) has CFI over 0.95.

Tucker–Lewis Fit Index (TLI) is quite similar as CFI (according to lecture slides), but it includes a penalty for too much complexity. Well fitting models have TLI close to 1. I think only model 1 TLI is close to number one, the others are smaller than this. However, it shouldn't be caused by complexity, because models 2 and 3 are simpler.

"Absolute misfit indices", RMSEA and SRMR depend only on how well the model fits the data.These decrease as the fit improves.
RMSEA (Root Mean Square Error Of Approximation): values less than 0.05 indicates good fit, and values greater that 0.1 indicates poor fit. Model 1 has RMSEA of 0.034 with 90 percent upper C.I of 0.062 (and p-value of 0.556). This indicates a goof fit. Even the lover 90 percent C.I. for both other models are greater than 0.1 indicating poor fit.

SRMR (Standardized Root Mean Square Residual) represents average residual value of the fit and according to lecture slides, a well fitting model has SRMR less than 0.05. Here Model 1 has SRMR of 0.048 indicating a good fit, and again the two other models have a larger number, SRMR over 0.1 indicating poor fit.

Conclusion: Model 1 fits well, even if the chi-square test refers to something else. Othes two models are poor fits.

What can you say about the parameter estimates?

Obviously parameter estimates should exhibit the correct sign and size. However, the content of the measured variables was not clear to me by browsing Byrne's book, so it is a bit difficult to conclude what would be the right direction and size.

I can still look at the statistical significance of the estimates and the size of the standard errors. The lecture slides said that the estimates should not be excessively large or small. I do love these expressions: they don't mean anything unless you know *what is* large or small.

Table 4: Model 1: Parameter estimates

lhs	rhs	est	se	z	pvalue
GSC	SDQ2N01	1.000	0.000	NA	NA
GSC	SDQ2N13	1.083	0.154	7.044	0
GSC	SDQ2N25	0.851	0.132	6.455	0
GSC	SDQ2N37	0.934	0.131	7.131	0
ASC	SDQ2N04	1.000	0.000	NA	NA
ASC	SDQ2N16	1.279	0.150	8.520	0
ASC	SDQ2N28	1.247	0.154	8.097	0
ASC	SDQ2N40	1.259	0.156	8.048	0
ESC	SDQ2N10	1.000	0.000	NA	NA
ESC	SDQ2N22	0.889	0.103	8.658	0
ESC	SDQ2N34	0.670	0.148	4.539	0
ESC	SDQ2N46	0.843	0.117	7.225	0
MSC	SDQ2N07	1.000	0.000	NA	NA
MSC	SDQ2N19	0.841	0.058	14.495	0
MSC	SDQ2N31	0.952	0.049	19.516	0
MSC	SDQ2N43	0.655	0.049	13.298	0

```r
# cfa1_sum <- summary(cfa1) # I dont want to prin the whole summary
cfa1_PE <- cfa1_sum$PE[1:16, c(1, 3, 5:8)]
cfa1_PE[, 3:6] <- round(cfa1_PE[, 3:6], 3)
cfa1_PE %>%
  kable(booktabs=T, align = "c", caption = "Model 1: Parameter estimates") %>%
  kable_styling(full_width = T)
```

In **Model 1** (Table 4), the estimates of all variables are statistically significant (p-values are close to zero). In my opinion, the standard errors are not disturbingly large or small.

```r
# cfa2_sum <- summary(cfa2) # I dont want to prin the whole summary
cfa2_PE <- cfa2_sum$PE[1:16, c(1, 3, 5:8)]
cfa2_PE[, 3:6] <- round(cfa2_PE[, 3:6], 3)
cfa2_PE %>%
  kable(booktabs=T, align = "c", caption = "Model 2: Parameter estimates") %>%
  kable_styling(full_width = T)
```

Model 2 (Table 5) has one variable (SDQ2N34) whose estimate is not statistically significant (p-value is 0.685). The standard errors are larger than in Model 1. However, I don't think they are disturbingly large or small.

```r
# cfa3_sum <- summary(cfa3) # I don't want to print the whole summary
cfa3_PE <- cfa3_sum$PE[1:16, c(1, 3, 5:8)]
cfa3_PE[, 3:6] <- round(cfa3_PE[, 3:6], 3)
cfa3_PE %>%
  kable(booktabs=T, align = "c", caption = "Model 3: Parameter estimates") %>%
  kable_styling(full_width = T)
```

Model 3 (Table 6) also has one variable (SDQ2N34) whose estimate is not statistically significant (p-value is 0.415). The standard errors are clearly larger than in Model 1. Should I be disturbed by these already?

Table 5: Model 2: Parameter estimates

lhs	rhs	est	se	z	pvalue
GSC	SDQ2N01	1.000	0.000	NA	NA
GSC	SDQ2N13	1.048	0.151	6.930	0.000
GSC	SDQ2N25	0.860	0.131	6.542	0.000
GSC	SDQ2N37	0.890	0.128	6.957	0.000
ASC	SDQ2N04	1.000	0.000	NA	NA
ASC	SDQ2N16	1.263	0.170	7.440	0.000
ASC	SDQ2N28	1.276	0.177	7.221	0.000
ASC	SDQ2N40	1.235	0.176	7.026	0.000
ASC	SDQ2N10	0.581	0.123	4.736	0.000
ASC	SDQ2N22	0.558	0.117	4.786	0.000
ASC	SDQ2N34	0.065	0.161	0.406	0.685
ASC	SDQ2N46	0.514	0.132	3.885	0.000
ASC	SDQ2N07	2.069	0.262	7.885	0.000
ASC	SDQ2N19	1.871	0.242	7.721	0.000
ASC	SDQ2N31	2.021	0.247	8.192	0.000
ASC	SDQ2N43	1.442	0.193	7.481	0.000

Table 6: Model 3: Parameter estimates

lhs	rhs	est	se	z	pvalue
ASC	SDQ2N01	1.000	0.000	NA	NA
ASC	SDQ2N13	1.158	0.247	4.690	0.000
ASC	SDQ2N25	0.903	0.209	4.330	0.000
ASC	SDQ2N37	1.126	0.224	5.018	0.000
ASC	SDQ2N04	1.407	0.278	5.063	0.000
ASC	SDQ2N16	1.772	0.310	5.716	0.000
ASC	SDQ2N28	1.775	0.317	5.605	0.000
ASC	SDQ2N40	1.744	0.315	5.541	0.000
ASC	SDQ2N10	0.859	0.197	4.362	0.000
ASC	SDQ2N22	0.816	0.187	4.371	0.000
ASC	SDQ2N34	0.181	0.222	0.815	0.415
ASC	SDQ2N46	0.756	0.202	3.732	0.000
ASC	SDQ2N07	2.743	0.471	5.826	0.000
ASC	SDQ2N19	2.505	0.434	5.768	0.000
ASC	SDQ2N31	2.711	0.454	5.970	0.000
ASC	SDQ2N43	1.929	0.341	5.659	0.000

However, I already decided that model 3 is not a good fit, so the possibly large standard errors are not a problem.

— That's all Folks —