

Psychometrics Assignment1

Kyuri Park 5439043

April 19 2022

```
# load data
dat <- read.table("Assign1.dat", header=T)
head(dat) # 728 examinees on 18 items

## specify the model: standard confirmatory three-factor model
model <- "f1 =~ x1 + x2 + x7 + x10 + x13 + x16
f2 =~ x3 + x4 + x8 + x11 + x14 + x17
f3 =~ x5 + x6 + x9 + x12 + x15 + x18"

## fit a three factor model (using unit variance identification)
mod <- cfa(model, std.lv=T, data = dat)
summary(mod, fit.measures = T, standardized=T)
```

Note: The model output (summary) can be found in the *Appendix* at the end of the document.

1. Does the model fit to the data?

Yes, the model fits to the data well given that:

- $\chi^2(132) = 141.45, p = 0.27$: since p-value is larger than 0.05, the model cannot be rejected. There is no statistical evidence against the null hypothesis (our three factor model).
- $CFI = 0.998$: CFI is above 0.95, which indicates a good fit.
- $TLI = 0.998$: TLI is above 0.95, which indicates a good fit.
- $RMSEA = 0.01$: RMSEA is below 0.05, which indicates a good fit.
- $SRMR = 0.024$: SRMR is below 0.08, which indicates a good fit.

2. Are the latent variables measured independent?

No, as seen below in the correlation matrix of the latent factors, they are not independent.

The correlation between the latent factors are fairly high (e.g., $cor(f1, f2) = 0.64$, $cor(f2, f3) = 0.49$). In addition, in the model summary, it is shown that the correlations between the latent factors are all significant (all $p < .001$; see the model output in *Appendix*), suggesting that the latent factors are not independent of each other.

```
## check the correlations between the latent factors
lavInspect(mod, "cor.lv") # same as psi as it the latent variance is set to 1
```

```
##      f1      f2      f3
## f1 1.000
## f2 0.644 1.000
## f3 0.264 0.489 1.000
```

3. What is the estimate of the total variance explained?

The estimated proportion of total variance of the item scores explained by the common factors is 0.543. It indicates that about 54% of the total variance is explained by the three common factors.

```
## extract the estimated uniqueness
theta <- inspect(mod, "est")$theta

## extract model implied covariance matrix
sigma <- fitted(mod)$cov

## compute the estimate of total variance explained by common factors
1 - (tr(theta)/tr(sigma))
```

```
## [1] 0.5433328
```

4. Give the estimates of the validities of the three unweighted subtest scores as measures of the latent variables.

The following validity estimates are obtained for S_1 , S_2 , and S_3 :

	C_1	C_2	C_3
S_1	0.913	0.379	0.063
S_2	0.348	0.838	0.201
S_3	0.058	0.200	0.836

```
## b: vector of weights
b1 <- matrix(c(rep(1,6),rep(0,12)), nrow=1)
b2 <- matrix(c(rep(0,6),rep(1,6), rep(0,6)), nrow=1)
```

```

b3 <- matrix(c(rep(0,12),rep(1,6)), nrow=1)

## lambda: estimated factor loading matrix
lambda <- inspect(mod, "est")$lambda

## psi: estimated covariance matrix of common factors
psi <- inspect(mod, "est")$psi

## var.C: variance of common factors
var.C1 <- diag(psi)[1]
var.C2 <- diag(psi)[2]
var.C3 <- diag(psi)[3]

## e: q by q identity matrix (q = number of common factors)
e1 <- as.matrix(diag(3)[,1])
e2 <- as.matrix(diag(3)[,2])
e3 <- as.matrix(diag(3)[,3])

## create a storage for the validity estimates of subtest scores
validity <- matrix(NA, 3, 3)
colnames(validity) <- c("C1", "C2", "C3")
rownames(validity) <- c("S1", "S2", "S3")

## estimate validities of subtest scores
validity[1,1] <- (b1 %%% lambda %%% psi %%% e1)^2 / (b1%%sigma %%% t(b1)*var.C1)
validity[1,2] <- (b1 %%% lambda %%% psi %%% e2)^2 / (b1%%sigma %%% t(b1)*var.C2)
validity[1,3] <- (b1 %%% lambda %%% psi %%% e3)^2 / (b1%%sigma %%% t(b1)*var.C3)

validity[2,1] <- (b2 %%% lambda %%% psi %%% e1)^2 / (b2%%sigma %%% t(b2)*var.C1)
validity[2,2] <- (b2 %%% lambda %%% psi %%% e2)^2 / (b2%%sigma %%% t(b2)*var.C2)
validity[2,3] <- (b2 %%% lambda %%% psi %%% e3)^2 / (b2%%sigma %%% t(b2)*var.C3)

validity[3,1] <- (b3 %%% lambda %%% psi %%% e1)^2 / (b3%%sigma %%% t(b3)*var.C1)
validity[3,2] <- (b3 %%% lambda %%% psi %%% e2)^2 / (b3%%sigma %%% t(b3)*var.C2)
validity[3,3] <- (b3 %%% lambda %%% psi %%% e3)^2 / (b3%%sigma %%% t(b3)*var.C3)

## show the validity estimates of subtest scores
round(validity, 3)

##      C1    C2    C3
## S1 0.913 0.379 0.063
## S2 0.348 0.838 0.201
## S3 0.058 0.200 0.836

```

5. Give the estimates of the communalities of the three unweighted subtest scores. What do you notice when you compare these to the corresponding validity estimates? Does that make sense? Explain.

The estimates of communalities of the unweighted subtest scores are shown as below. Each of the communality estimate is the same as corresponding validity estimate above (diagonal elements), and this makes sense because here we have specified a standard confirmatory factor model. Since there is only one common factor underlying for each subtest, the validity estimates coincide with the communality estimates.

	S_1	S_2	S_3
Communality	0.913	0.838	0.836

```
## create a storage for the validity estimates of subtest scores
communality <- matrix(NA, 1, 3) # storage for the following communality estimates
colnames(communality) <- c("S1", "S2", "S3")

## estimate communalities of subtest scores
communality[,1] <- 1 - (b1 %*% theta %*% t(b1))/(b1 %*% sigma %*% t(b1))
communality[,2] <- 1 - (b2 %*% theta %*% t(b2))/(b2 %*% sigma %*% t(b2))
communality[,3] <- 1 - (b3 %*% theta %*% t(b3))/(b3 %*% sigma %*% t(b3))

## show the validity estimates of subtest scores
round(communality, 3)
```

```
##           S1      S2      S3
## [1,] 0.913 0.838 0.836
```

6. Give the estimates of the validities of Thurstone factor scores as measures of the latent variables.

The following validity estimates are obtained for Thurstone factor scores F_1 , F_2 and F_3 :

	C_1	C_2	C_3
F_1	0.923	0.434	0.075
F_2	0.461	0.868	0.262
F_3	0.080	0.263	0.867

```
## estimate Thurstone's factor scores
b_Thurstone <- as.matrix(psi %*% t(lambda) %*% solve(sigma))
b_ThurstoneF1 <- b_Thurstone[1,] # extract the first row
b_ThurstoneF2 <- b_Thurstone[2,] # extract the second row
b_ThurstoneF3 <- b_Thurstone[3,] # extract the third row

Thurstone.val <- matrix(NA, 3,3) # storage for following validity estimates
colnames(Thurstone.val) <- c("C1", "C2", "C3")
```

```

rownames(Thurstone.val) <- c("F1", "F2", "F3")

## estimate the validity of Thurstone factor scores:
Thurstone.val[1,1] <- (t(b_ThurstoneF1) %*% lambda %*% psi %*% e1)^2 / (t(b_ThurstoneF1)
%*% sigma %*% b_ThurstoneF1 * var.C1)
Thurstone.val[1,2] <- (t(b_ThurstoneF1) %*% lambda %*% psi %*% e2)^2 / (t(b_ThurstoneF1)
%*% sigma %*% b_ThurstoneF1 * var.C2)
Thurstone.val[1,3] <- (t(b_ThurstoneF1) %*% lambda %*% psi %*% e3)^2 / (t(b_ThurstoneF1)
%*% sigma %*% b_ThurstoneF1 * var.C3)
Thurstone.val[2,1] <- (t(b_ThurstoneF2) %*% lambda %*% psi %*% e1)^2 / (t(b_ThurstoneF2)
%*% sigma %*% b_ThurstoneF2 * var.C1)
Thurstone.val[2,2] <- (t(b_ThurstoneF2) %*% lambda %*% psi %*% e2)^2 / (t(b_ThurstoneF2)
%*% sigma %*% b_ThurstoneF2 * var.C2)
Thurstone.val[2,3] <- (t(b_ThurstoneF2) %*% lambda %*% psi %*% e3)^2 / (t(b_ThurstoneF2)
%*% sigma %*% b_ThurstoneF2 * var.C3)
Thurstone.val[3,1] <- (t(b_ThurstoneF3) %*% lambda %*% psi %*% e1)^2 / (t(b_ThurstoneF3)
%*% sigma %*% b_ThurstoneF3 * var.C1)
Thurstone.val[3,2] <- (t(b_ThurstoneF3) %*% lambda %*% psi %*% e2)^2 / (t(b_ThurstoneF3)
%*% sigma %*% b_ThurstoneF3 * var.C2)
Thurstone.val[3,3] <- (t(b_ThurstoneF3) %*% lambda %*% psi %*% e3)^2 / (t(b_ThurstoneF3)
%*% sigma %*% b_ThurstoneF3 * var.C3)
round(Thurstone.val, 3)

```

```

##      C1    C2    C3
## F1 0.923 0.434 0.075
## F2 0.461 0.868 0.262
## F3 0.080 0.263 0.867

```

7. Calculate estimates of the correlations among the Thurstone factor scores.

The estimated correlations among the Thurstone factor scores are given in the matrix:

$$\begin{bmatrix} 1.000 & 0.707 & 0.295 \\ 0.707 & 1.000 & 0.550 \\ 0.295 & 0.550 & 1.000 \end{bmatrix}$$

```

## center the data
centered_dat <- apply(dat, 2, function(y) y - mean(y))
## match the order of item in data as it is in the Thurstone weight matrix
centered_dat <- centered_dat[,colnames(b_Thurstone)]
## compute the Thurstone factor scores
F_Thurstone <- centered_dat %*% t(b_Thurstone)

```

```
## correlation matrix
round(cor(F_Thurstone), 3)
```

```
##      f1    f2    f3
## f1 1.000 0.707 0.295
## f2 0.707 1.000 0.550
## f3 0.295 0.550 1.000
```

8. Calculate the communality of the unweighted total test score. What does this estimate tell you about the reliability of the unweighted total test score?

The estimated communality of the unweighted total test score (Y) is 0.93, which is high. Given the fact that the communality of Y is a lower bound to the reliability of Y , it can be concluded that the reliability of Y is also high (greater than or equal to 0.93).

```
## b: vector of weight (set to 1) --> unweighted total test score
b <- matrix(1, nrow=1, ncol=18)

## estimate the communality of unweighted total test score
comm.total <- 1 - (b %*% theta %*% t(b))/(b %*% sigma %*% t(b))
drop(round(comm.total, 3))
```

```
## [1] 0.929
```

9. Assuming essentially parallel multivariate normal item scores, give the estimate of Cronbach's alpha for the unweighted total test score and test the hypothesis that Cronbach's alpha is .8 against a one-sided alternative.

The estimate of Cronbach's alpha for the unweighted total test score is 0.8943.

Kristof's unbiased estimate of alpha for the unweighted total test score is 0.8946.

The null hypothesis that Cronbach's alpha is equal to 0.8 is rejected as $p < .001$, meaning that it significantly deviates from the hypothesized value of 0.8. Thus, it seems not reasonable to assume that the true reliability is 0.8 in the population .

```
## estimate the Cronbach's alpha
k <- 18 # number of items
n <- nrow(dat) # number of examinees = 728
S <- cov(dat)
alpha <- k*(sum(S)-sum(diag(S)))/(k-1)/sum(S)
cat("estimate of Cronbach's alpha = ", round(alpha, 4))
```

```
## estimate of Cronbach's alpha = 0.8943
```

```
## Kristof's unbiased alpha
unbiased_alpha <- (2 / (n-1)) + ((n-3)/(n-1)*alpha)
cat("Kristof's unbiased estimate of alpha = ", round(unbiased_alpha,4))
```

```
## Kristof's unbiased estimate of alpha = 0.8946
```

```
## F-statistic: hypothesized alpha value = 0.8
F <- (n-3)/(n-1) * (1 - 0.8) / (1-unbiased_alpha)
```

```
## hypothesis testing: obtain the p-value
pval <- 1 - pf(F, df1 = n-1, df2 = (n-1)*(k-1))
cat("p-value = ", pval)
```

```
## p-value = 0
```

10. Give an overall conclusion and interpretation.

Given the initial hypothesis on the factor structure, a standard confirmatory three factor model is fitted to the data. Based on the goodness of fit test and other fit indices, it is concluded that the three factor model fits well to the data.

As shown in the output (see *Appendix*), all the factor loadings are significant, which means that the indicators are significantly related to the common factors. In addition, the output shows that the three common factors are significantly correlated, implying that they are not measured independently. However, the latent correlations between the factors are deemed to be not too strong to cast a substantial doubt on the three-factor structure.

The total variance explained is estimated to be 0.54, which is not very high but not low either. About 54% of total variance can be explained by the three common factors, and I'd say this is quite a considerable amount and hence, the 18-item test is considered to have an acceptable construct validity.

The estimated validity coefficients are high (all above 0.8 for the diagonal), thus it is concluded that the subtests measure what they are supposed to measure (hypothesized corresponding factor). Using Thurstone factor scores gives a bit higher validity estimates compared to using the subtest scores, thus I'd use Thurstone factor scores instead of the subtest scores in this case. Additionally, the estimated communality of the unweighted total test score is high (0.93), which implies that the reliability of the unweighted total test score is high as well.

All in all, the hypothesized three factor model does seem to be supported by the observed data. The model fits good, the construct validity of the test is deemed to be acceptable, the estimated validity coefficients are high (all above 0.85 based on the Thurstone factor scores), and the reliability of the total test score is also high (≥ 0.93).

Appendix

Three-factor model output

```
## lavaan 0.6-10 ended normally after 40 iterations
##
## Estimator ML
## Optimization method NLMINB
## Number of model parameters 39
##
## Number of observations 728
##
## Model Test User Model:
##
## Test statistic 141.453
## Degrees of freedom 132
## P-value (Chi-square) 0.271
##
## Model Test Baseline Model:
##
## Test statistic 6180.043
## Degrees of freedom 153
## P-value 0.000
##
## User Model versus Baseline Model:
##
## Comparative Fit Index (CFI) 0.998
## Tucker-Lewis Index (TLI) 0.998
##
## Loglikelihood and Information Criteria:
##
## Loglikelihood user model (H0) -11423.406
## Loglikelihood unrestricted model (H1) -11352.679
##
## Akaike (AIC) 22924.811
## Bayesian (BIC) 23103.833
## Sample-size adjusted Bayesian (BIC) 22979.996
##
## Root Mean Square Error of Approximation:
##
## RMSEA 0.010
## 90 Percent confidence interval - lower 0.000
## 90 Percent confidence interval - upper 0.021
## P-value RMSEA <= 0.05 1.000
##
## Standardized Root Mean Square Residual:
```



```

##
## SRMR 0.024
##
## Parameter Estimates:
##
## Standard errors Standard
## Information Expected
## Information saturated (h1) model Structured
##
## Latent Variables:
## Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## f1 =~
## x1 0.522 0.022 23.475 0.000 0.522 0.757
## x2 0.735 0.030 24.800 0.000 0.735 0.786
## x7 0.488 0.022 21.732 0.000 0.488 0.716
## x10 0.699 0.024 29.152 0.000 0.699 0.873
## x13 0.823 0.029 28.165 0.000 0.823 0.854
## x16 0.724 0.030 24.343 0.000 0.724 0.776
## f2 =~
## x3 0.567 0.029 19.652 0.000 0.567 0.680
## x4 0.483 0.025 19.680 0.000 0.483 0.681
## x8 0.485 0.025 19.535 0.000 0.485 0.677
## x11 0.457 0.026 17.617 0.000 0.457 0.624
## x14 0.424 0.018 23.537 0.000 0.424 0.776
## x17 0.430 0.021 20.205 0.000 0.430 0.694
## f3 =~
## x5 0.448 0.026 17.418 0.000 0.448 0.619
## x6 0.632 0.026 24.742 0.000 0.632 0.806
## x9 0.355 0.026 13.737 0.000 0.355 0.508
## x12 0.595 0.023 25.491 0.000 0.595 0.823
## x15 0.324 0.018 18.435 0.000 0.324 0.648
## x18 0.430 0.025 17.131 0.000 0.430 0.611
##
## Covariances:
## Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## f1 ~~
## f2 0.644 0.027 24.067 0.000 0.644 0.644
## f3 0.264 0.039 6.743 0.000 0.264 0.264
## f2 ~~
## f3 0.489 0.035 14.146 0.000 0.489 0.489
##
## Variances:
## Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .x1 0.203 0.012 16.909 0.000 0.203 0.427
## .x2 0.334 0.020 16.454 0.000 0.334 0.382
## .x7 0.226 0.013 17.381 0.000 0.226 0.487

```

##	.x10	0.153	0.011	13.851	0.000	0.153	0.238
##	.x13	0.251	0.017	14.658	0.000	0.251	0.270
##	.x16	0.345	0.021	16.623	0.000	0.345	0.397
##	.x3	0.374	0.023	16.556	0.000	0.374	0.538
##	.x4	0.270	0.016	16.546	0.000	0.270	0.537
##	.x8	0.278	0.017	16.599	0.000	0.278	0.542
##	.x11	0.327	0.019	17.210	0.000	0.327	0.610
##	.x14	0.119	0.008	14.616	0.000	0.119	0.398
##	.x17	0.198	0.012	16.344	0.000	0.198	0.518
##	.x5	0.322	0.019	17.220	0.000	0.322	0.616
##	.x6	0.215	0.016	13.355	0.000	0.215	0.350
##	.x9	0.361	0.020	18.048	0.000	0.361	0.742
##	.x12	0.169	0.013	12.632	0.000	0.169	0.322
##	.x15	0.145	0.009	16.912	0.000	0.145	0.580
##	.x18	0.310	0.018	17.300	0.000	0.310	0.626
##	f1	1.000				1.000	1.000
##	f2	1.000				1.000	1.000
##	f3	1.000				1.000	1.000