Solutions to Lab 8

Multivariate Statistics with R

As promised last week, in this lab, we will delve deeper into fit indices and model comparison. So, without further ado:

Task 1: Run the first example in `umxRAM()` documentation, just like you did last week.

```
### ANSWER ###

myData <- mxData(cov(mtcars[ , c(1, 3, 6)]), "cov", numObs = nrow(mtcars))

m1 <- umxRAM("tim", data = myData,
             umxPath(c("wt", "disp"), to = "mpg"),
             umxPath("wt", with = "disp"),
             umxPath(var = c("wt", "disp", "mpg")))
```

```
## [1] "<U+03C7>²(0) = -0.05, p = 1.000; CFI = 1.001; TLI = 1; RMSEA = 0"
```

Task 2: Get a summary of the model using the `umx` helper function.

```
### ANSWER ###

umxSummary(m1)
```

```
##
##
## |name           | Estimate|      SE|
## |:--------------|--------:|-------:|
## |disp_to_mpg    |    -0.02|    0.01|
## |wt_to_mpg      |    -3.35|    1.11|
## |mpg_with_mpg   |     7.71|    1.93|
## |disp_with_disp | 14880.75| 3720.16|
## |disp_with_wt   |   104.32|   27.77|
## |wt_with_wt     |     0.93|    0.23|
## [1] "<U+03C7>²(0) = -0.05, p = 1.000; CFI = 1.001; TLI = 1; RMSEA = 0"
```

Question 2.1: Does it fit well?

```
### ANSWER ###
```

As discussed last week, yes, this is a perfect fit because the model is saturated.

Task 3: Get a `summary()` of the model.

```
### ANSWER ###

summary(m1)
```

```
## Summary of tim
##
## free parameters:
##              name matrix  row  col      Estimate    Std.Error A lbound
## 1    disp_to_mpg      A  mpg disp -1.772516e-02 8.748828e-03
## 2      wt_to_mpg      A  mpg   wt -3.350778e+00 1.108187e+00
```

```
## 3   mpg_with_mpg      S  mpg  mpg  7.708730e+00 1.927117e+00 !      0
## 4 disp_with_disp      S disp disp  1.488075e+04 3.720157e+03        0
## 5   disp_with_wt      S disp   wt  1.043185e+02 2.777299e+01
## 6     wt_with_wt      S   wt   wt  9.274526e-01 2.318590e-01        0
##   ubound
## 1
## 2
## 3
## 4
## 5
## 6
##
## Model Statistics:
##               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             6                      0              416.6821
##    Saturated:             6                      0              416.7300
## Independence:             3                      3              515.0320
## Number of observations/statistics: 32/6
##
## chi-square:  <U+03C7>² ( df=0 ) = -0.04787503,  p = 1
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:    -0.04787503              11.95212                       NA
## BIC:    -0.04787503              20.74654                 2.041965
## CFI: 1.000502
## TLI: 1   (also known as NNFI)
## RMSEA:  0  [95% CI (NA, NA)]
## Prob(RMSEA <= 0.05): NA
## timestamp: 2017-11-01 13:41:06
## Wall clock time: 0.5021999 secs
## optimizer:  CSOLNP
## OpenMx version number: 2.7.18
## Need help?  See help(mxSummary)
```

Question 3.1: What fit statistics can you see?

```
### ANSWER ###
```

- χ^2 (Chi-squared) measure of fit based on $-2LL$
- *AIC*, Akaike Information Criterion (comparative model fit penalised for degrees of freedom)
- *BIC*, Bayes Information Criterion
- *CFI*, Comparative fit Index
- *TLI*, Tucker-Lewis index
- *RMSEA*, Root mean square error of approximation, some call it 'Ramsey'

Task 4: Inspect the model fit.

Question 4.1: Is the fit of the model good according to RMSEA and TLI?

```
### ANSWER ###
```

Yes, according to the conventional thresholds (see below), the model fits very well.

Question 4.2: What are conventional criteria for good fit?

```
### ANSWER ###
```

RMSEA $\leq .05$

TLI $\geq .95$

Question 4.3: Can you tell from the AIC if fit is good?

```
### ANSWER ###
```

No, AIC is a comparative measure so it can only tell you whether your model fits better or worse than some other model. The value of AIC in and of itself does not indicate whether or not the fit is good.

Task 5: Look up the formula for AIC in the `summary.MxModel()` documentation.

Question 5.1: Explain this to a lab-mate.

```
### ANSWER ###
```

The formula is

$$AIC = -2LL + 2 \times N_{param}$$

, where $-2LL$ is the model fit in terms of $-2\times$log-likelihood and N_{param} is the number of free parameters. Because we're adding $2\times N_{param}$, we are penalising the given model for complexity (more complex models estimate more free parameters). In the case of `m1`, the AIC value is `AIC(m1)` = `summary(m1)$Minus2LogLikelihood + 2 * summary(m1)$estimatedParameters` = 428.6821497.

Task 6: Look up the formula for RMSEA on the internet.

```
### ANSWER ###
```

The formula is

$$\frac{\sqrt{\chi^2 - df}}{\sqrt{df \times (N-1)}}$$

Question 6.1: What are the key parameters?

```
### ANSWER ###
```

- χ^2 is the difference in $-2LL$ of the model versus the saturated model
- *df* are the model degrees of freedom
- N is the sample size.

Question 6.2: What makes RMSEA get smaller?

```
### ANSWER ###
```

RMSEA gets smaller as:

a. the numerator ($\sqrt{\chi^2 - df}$) gets smaller, which happens as a.1. the fit of the model (at a given level of complexity) improves, *i.e.,* the χ^2 gets smaller, or a.2. the model gets simpler (at a given level of fit), *i.e., df* gets bigger.

b. the denominator ($\sqrt{df \times (N-1)}$) gets bigger, which happens as… see below.

Question 6.3: Plug in some values and see…

```
### ANSWER ###

# You can use a functun such as this to play with different values
rmsea <- function(chisq, df, n) {
  return(sqrt(chisq - df)/sqrt(chisq * (n - 1)))
}

# e.g.
rmsea(25.43, 10, 300)
```

```
## [1] 0.04504786
```

Figure 1 shows a visual representation of the relationship between χ^2 (between 1 and 50) and model *df* (1 - 20) on the one hand and RMSEA on the other. The flat blue section reflects the fact that if $df > \chi^2$, RMSEA is set to be zero, instead of negative.

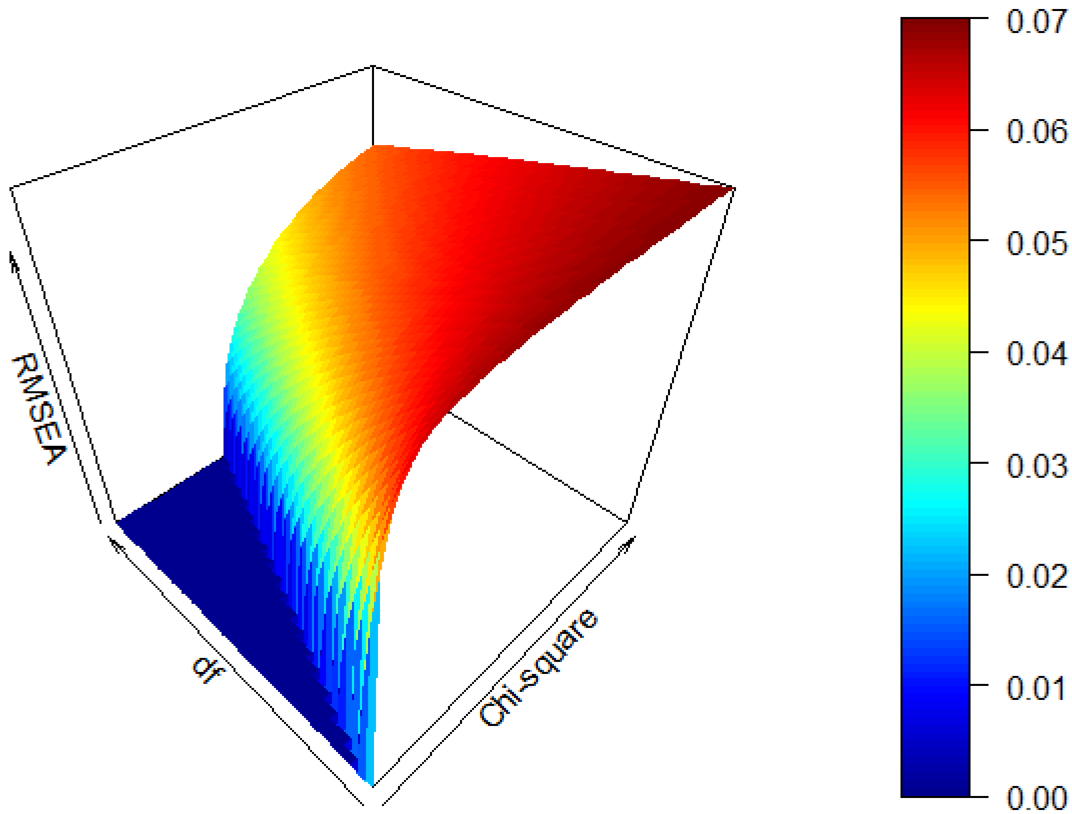


Figure 1: RMSEA as function of χ^2 and model *df* (at $N = 200$)

Question 6.4: What makes the denominator get bigger?

```
### ANSWER ###
```

Two things:

- larger sample size (N)
- simpler model (more *df*)

Task 7: get the mxRefModels for your model `m1`

```
### ANSWER ###

ref <- mxRefModels(m1)
```

```
[omitting printout to save space]
```

Question 7.1: What does mxRefModels return?

```
### ANSWER ###
```

A list of two models: `Saturated` (the best possible) and `Independence` (the worst possible).

Question 7.2: What are these two reference models?

```
### ANSWER ###
```

`Saturated` model is a model with all possible paths included (all possible parameters estimated, in our case six). Therefore if has zero degrees of freedom. This model will reproduce the orginal variance-covariance matrix of the data perfectly and so has a perfect fit. Unfortunately, that also makes it useless for hypothesis testing.

`Independence` model is the null model. It assumes no relationships between the variables (*i.e.,* all the covariances are fixed to 0) and only estimates the variances of the variables provided (in this case 3 variables, hence 3 parameters and $6 - 3 = 3$ degrees of freedom).

Question 7.3: Why are they useful?

```
### ANSWER ###
```

They are used to derive some of the fit indices. For instance, the Tucker-Lewis Index is calculated as:

$$TLI = \frac{Fit_{null\ model} - Fit_{my\ model}}{Fit_{null\ model} - 1}$$

, where $Fit_x = \frac{\chi^2_x}{df_x}$.

Task 8: Run the example model `m1` given in `?mxRefModels`.

```
### ANSWER ###

data(demoOneFactor)
manifests <- names(demoOneFactor)
latents <- c("G")
```

```r
# A raw-data example where using mxRefModels adds fit indices
## Not run:
m1 <- mxModel("OneFactor", type = "RAM",
    manifestVars = manifests, latentVars = latents,
    mxPath(latents, to = manifests, values = diag(var(demoOneFactor))*.2),
    mxPath(manifests, arrows = 2, values = diag(var(demoOneFactor))*.8),
    mxPath(latents, arrows = 2, free = FALSE, values = 1),
    mxPath("one", to = latents, free = FALSE, values = 0),
    mxPath("one", to = manifests, values = 0),
    mxData(demoOneFactor, type = "raw")
)
m1 <- mxRun(m1)
```

Question 8.1: Produce `summary()` of `m1`.

```r
### ANSWER ###

summary(m1)
```

```
## Summary of OneFactor
##
## free parameters:
##                name matrix row col    Estimate   Std.Error A
## 1  OneFactor.A[1,6]      A  x1   G  0.39675454 0.015518929
## 2  OneFactor.A[2,6]      A  x2   G  0.50315689 0.018196349
## 3  OneFactor.A[3,6]      A  x3   G  0.57666356 0.020407861
## 4  OneFactor.A[4,6]      A  x4   G  0.70207014 0.023963841
## 5  OneFactor.A[5,6]      A  x5   G  0.79545293 0.026616616
## 6  OneFactor.S[1,1]      S  x1  x1  0.04073254 0.002804281
## 7  OneFactor.S[2,2]      S  x2  x2  0.03794390 0.002797372
## 8  OneFactor.S[3,3]      S  x3  x3  0.04074550 0.003142852
## 9  OneFactor.S[4,4]      S  x4  x4  0.03930825 0.003398648
## 10 OneFactor.S[5,5]      S  x5  x5  0.03621452 0.003667527
## 11 OneFactor.M[1,1]      M   1  x1 -0.04007965 0.019907021
## 12 OneFactor.M[1,2]      M   1  x2 -0.04584025 0.024129140
## 13 OneFactor.M[1,3]      M   1  x3 -0.05588405 0.027323335
## 14 OneFactor.M[1,4]      M   1  x4 -0.05581624 0.032625300
## 15 OneFactor.M[1,5]      M   1  x5 -0.07555257 0.036577381
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             15                   2485              934.096
##    Saturated:             20                   2480                   NA
## Independence:             10                   2490                   NA
## Number of observations/statistics: 500/2500
##
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -4035.904               964.096                       NA
## BIC:     -14509.205              1027.315                 979.7042
## To get additional fit indices, see help(mxRefModels)
## timestamp: 2017-11-01 13:41:12
## Wall clock time: 0.9224 secs
```

```
## optimizer:  CSOLNP
## OpenMx version number: 2.7.18
## Need help?  See help(mxSummary)
```

Question 8.2: No get another summary, this time providing `mxRefModels()` of the model to the `refModels` argument of `summary()`.

```
### ANSWER ###

summary(m1, refModels = mxRefModels(m1, run = TRUE))
```

```
## Summary of OneFactor
##
## free parameters:
##                  name matrix row col    Estimate   Std.Error A
## 1  OneFactor.A[1,6]      A  x1   G  0.39675454 0.015518929
## 2  OneFactor.A[2,6]      A  x2   G  0.50315689 0.018196349
## 3  OneFactor.A[3,6]      A  x3   G  0.57666356 0.020407861
## 4  OneFactor.A[4,6]      A  x4   G  0.70207014 0.023963841
## 5  OneFactor.A[5,6]      A  x5   G  0.79545293 0.026616616
## 6  OneFactor.S[1,1]      S  x1  x1  0.04073254 0.002804281
## 7  OneFactor.S[2,2]      S  x2  x2  0.03794390 0.002797372
## 8  OneFactor.S[3,3]      S  x3  x3  0.04074550 0.003142852
## 9  OneFactor.S[4,4]      S  x4  x4  0.03930825 0.003398648
## 10 OneFactor.S[5,5]      S  x5  x5  0.03621452 0.003667527
## 11 OneFactor.M[1,1]      M   1  x1 -0.04007965 0.019907021
## 12 OneFactor.M[1,2]      M   1  x2 -0.04584025 0.024129140
## 13 OneFactor.M[1,3]      M   1  x3 -0.05588405 0.027323335
## 14 OneFactor.M[1,4]      M   1  x4 -0.05581624 0.032625300
## 15 OneFactor.M[1,5]      M   1  x5 -0.07555257 0.036577381
##
## Model Statistics:
##                |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
##        Model:             15                   2485              934.0960
##    Saturated:             20                   2480              926.6972
## Independence:             10                   2490             4659.2219
## Number of observations/statistics: 500/2500
##
## chi-square:  <U+03C7>² ( df=5 ) = 7.3988,  p = 0.1926299
## Information Criteria:
##       |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
## AIC:      -4035.904               964.096                       NA
## BIC:     -14509.205              1027.315                 979.7042
## CFI: 0.9993556
## TLI: 0.9987112   (also known as NNFI)
## RMSEA:  0.03097612  [95% CI (0, 0.08145576)]
## Prob(RMSEA <= 0.05): 0.713215
## timestamp: 2017-11-01 13:41:12
## Wall clock time: 0.9224 secs
## optimizer:  CSOLNP
## OpenMx version number: 2.7.18
## Need help?  See help(mxSummary)
```

Question 8.3: What is the difference?

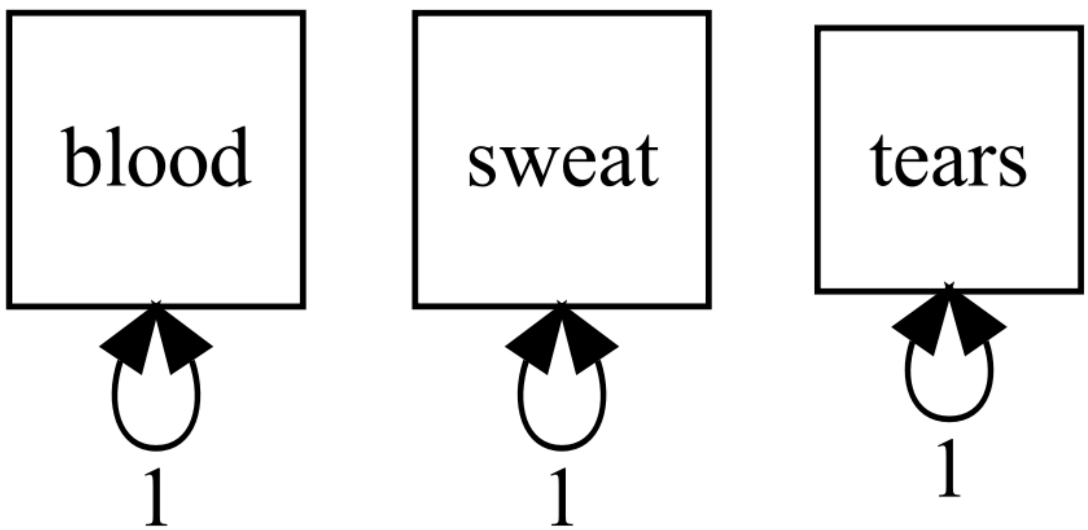
```
### ANSWER ###
```

First, in the `Model Statistics` section, we can now see the $-2LL$ values for the reference models.

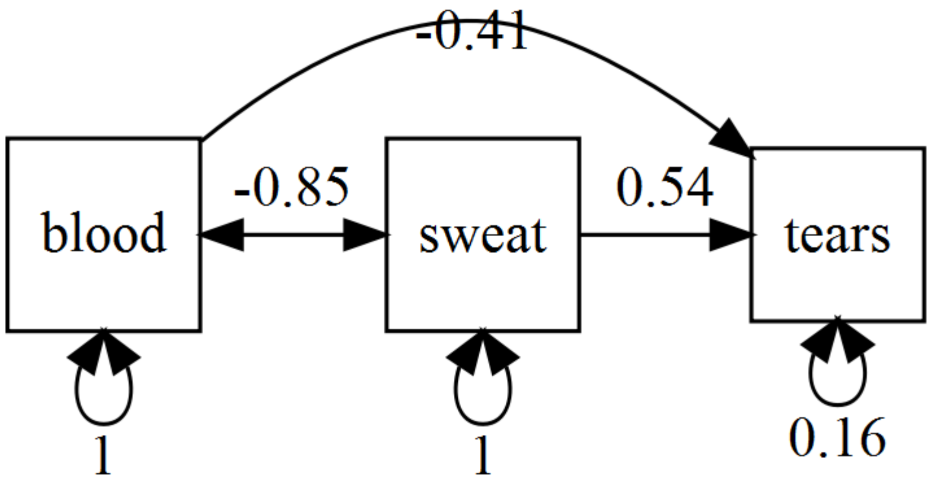
Next, the χ^2 test became available. This is a likelihood ratio test (think back to mixed-effects models) comparing our model `m1` to the saturated model. It is not significant ($p \geq .05$), meaning that `m1` does not fix significantly *worse* than the perfectly fitting model. Notice that the value of the χ^2 statistic corresponds to $-2LL_{m1} - -2LL_{saturated}$.

Finally, some mode fit indices now appear in the summary, namely CFI, TLI, and RMSEA. In addition, the latter comes with its own confidence interval and the probability of it being less than or equal to .05.

Task 9: Draw an independence model for three variables.



Task 10: Make it into saturated model for three variables.



Task 11: Open http://davidakenny.net/cm/fit.htm

Question 11.1: Try and figure out why the new statistics became available when the independence and saturated models became available.

```
### ANSWER ###
```

Simply put, they are needed! For example, the likelihood ratio test requires the $-2LL$ of the saturated model and the TLI needs the $-2LL$ of the null (independence) model.

Task 12: Take turns explaining to a lab-mate what optimisation does

That's it for this week. Well done!

Useful links

David Kenny's page

`umx` home page

`OpenMx` home page