Multivariate Statistics and Methodology using R

Confirmatory Factor Analysis

Aja Murray

Aja.Murray@ed.ac.uk

This Week

- Techniques
    - *Confirmatory Factor Analysis (CFA)*
- Key Functions
    - *cfa( ) from lavaan package*
- Reading
    - *lavaan tutorial: http://lavaan.ugent.be/tutorial/tutorial.pdf (sections 1-4)*
    - *lavaan paper: https://www.jstatsoft.org/article/view/v048i02*
    - *Confirmatory Factor Analysis chapter on Learn*

Learning Outcomes

- Know what it means to specify, estimate, and evaluate a CFA model
- Fit and interpret CFA models in R using the cfa( ) function
- Visualise CFA models using SEM diagrams

Overview of this lecture

- Introduction to CFA
- Model Specification
- Model Identification
- Model Estimation
- Model Evaluation
- Model Modification

Introduction to CFA

- Used to test a factor structure for a set of variables
- EFA is used when we have no fixed idea of our factor structure
- CFA is used to test a particular factor structure
- CFA tests how well our proposed factor structure fits the data
- Like EFA, CFA is a latent variable model
    - *observed variables serve as **indicators** of underlying latent factors*
- Unlike EFA, only specific loadings are included in the model

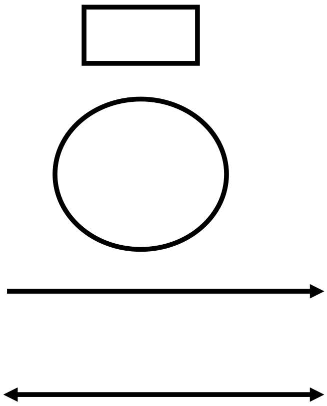
The variance-covariance matrix

- Our starting point for CFA is the variance-covariance matrix for our items
- CFA models represent these variances/covariances in terms of a set of latent factors

```
round(cov(agg.items),2)
```

```
##        item1 item2 item3 item4 item5 item6 item7 item8 item9 item10
## item1   0.98  0.56  0.47  0.44  0.59 -0.01  0.09  0.06  0.10   0.05
## item2   0.56  1.00  0.53  0.52  0.67  0.01  0.10  0.06  0.11   0.06
## item3   0.47  0.53  0.98  0.46  0.58  0.04  0.09  0.04  0.10   0.02
## item4   0.44  0.52  0.46  1.00  0.56  0.04  0.10  0.04  0.11   0.06
## item5   0.59  0.67  0.58  0.56  1.01  0.01  0.08  0.04  0.08   0.04
## item6  -0.01  0.01  0.04  0.04  0.01  1.03  0.58  0.60  0.43   0.45
## item7   0.09  0.10  0.09  0.10  0.08  0.58  0.98  0.81  0.58   0.60
## item8   0.06  0.06  0.04  0.04  0.04  0.60  0.81  1.02  0.61   0.64
## item9   0.10  0.11  0.10  0.11  0.08  0.43  0.58  0.61  0.98   0.45
## item10  0.05  0.06  0.02  0.06  0.04  0.45  0.60  0.64  0.45   1.01
```

SEM Diagrams



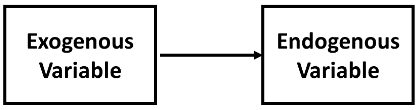
Observed variable

Latent variable

Regression path

(Co)variance

Exogenous versus endogeneous variables

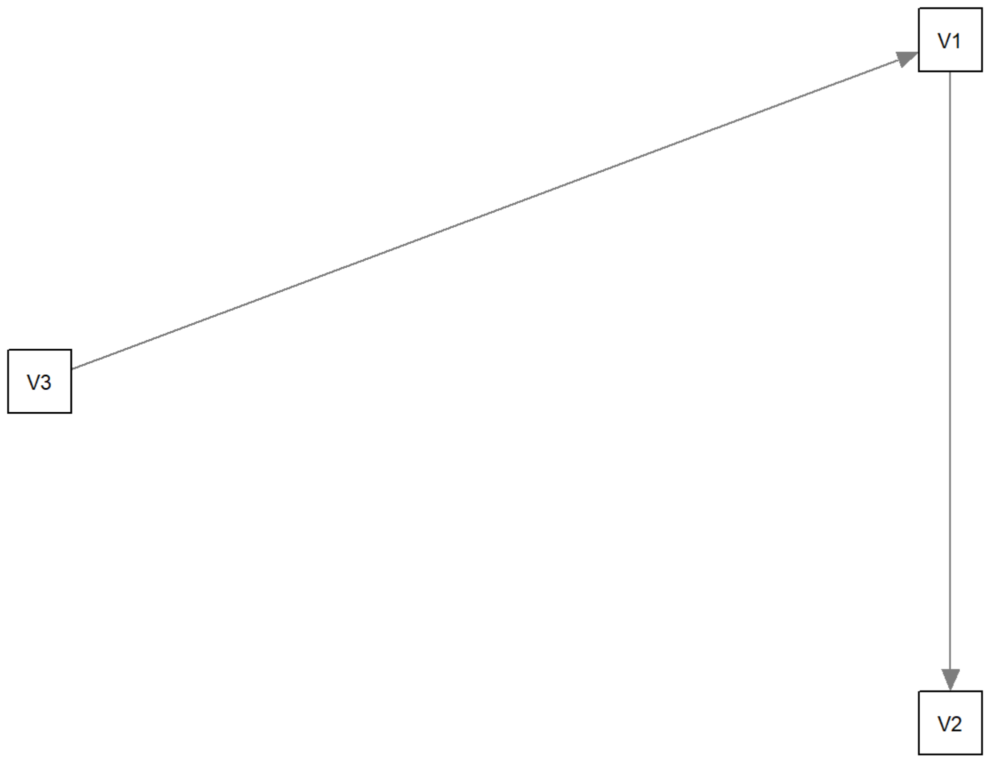


- **exogenous** variables receive input from no other variables
  - *they emanate but are not on the end of single-headed arrow paths*
  - *they are the 'independent variables' or 'predictors'*
- **endogenous** variables receive input from other variables
  - *they are on the end of single-headed arrow paths*
  - *they are the 'dependent variables' or 'outcomes'*
  - *they may also be predictors of other variables in the model*

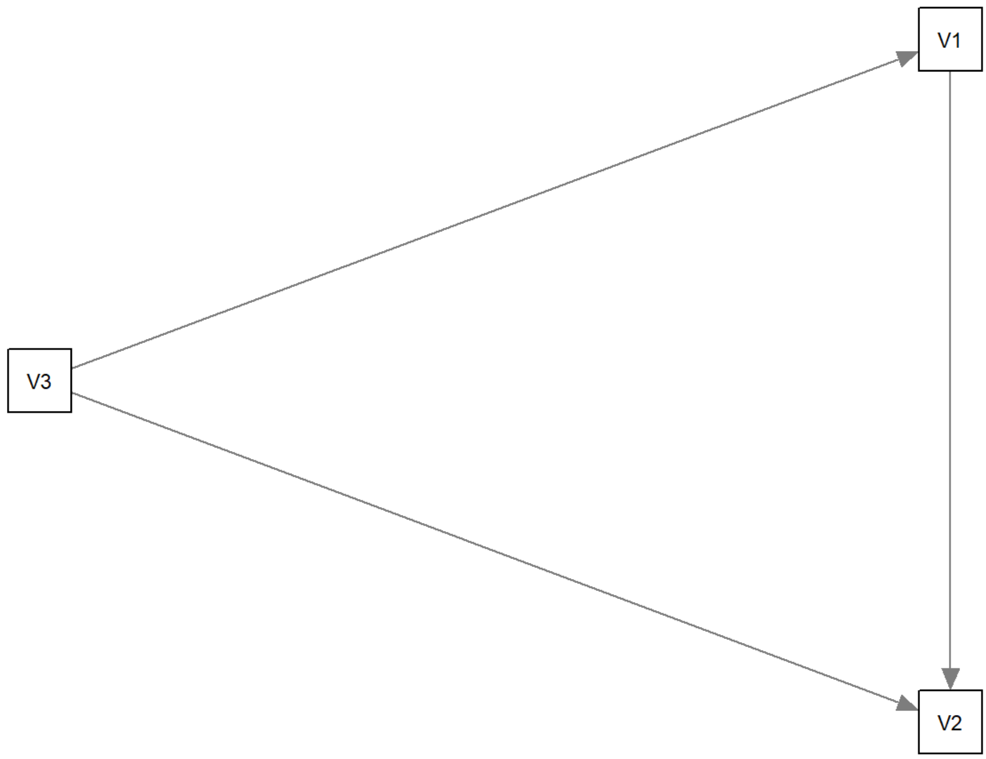
SEM diagram for a simple regression model

SEM diagram for a covariance

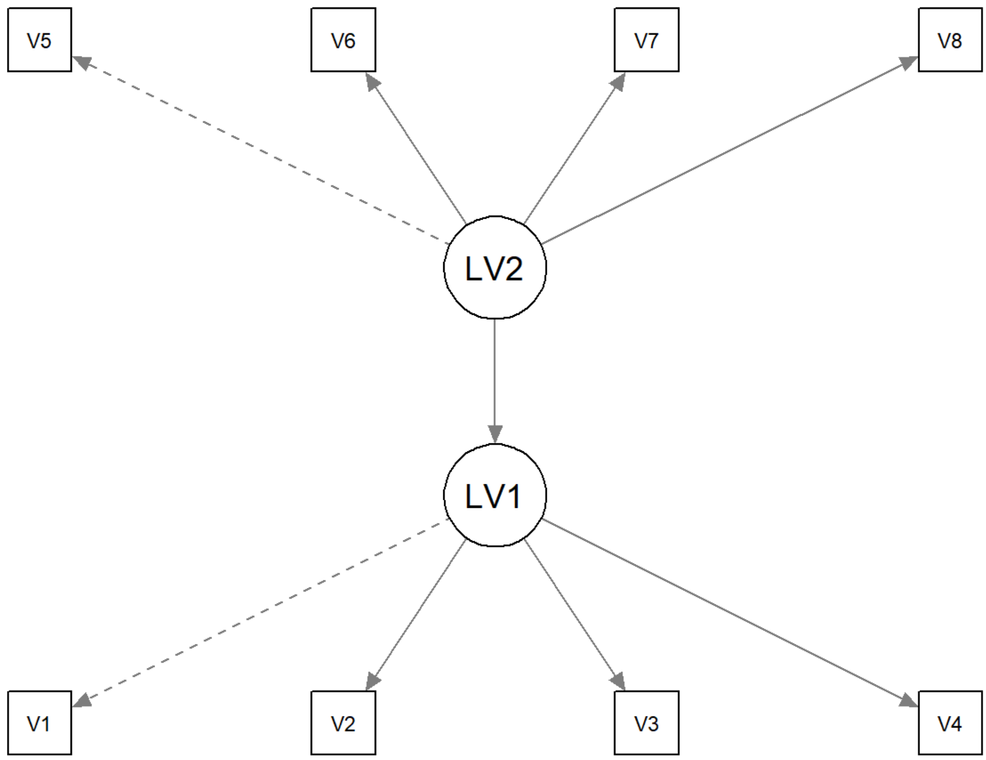
SEM diagram for a path analysis model



SEM diagram for another path analysis model



SEM diagram for a more complex model



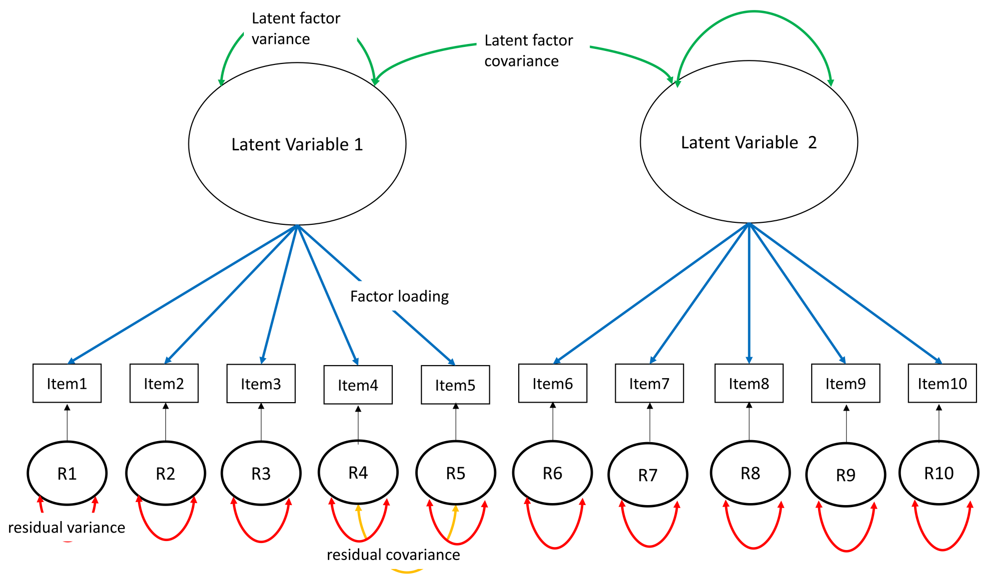
BREAK 1

- Time for a pause
- Quiz question:
  - *Which of these are differences between an EFA and a CFA:*
    - 1. EFA only involves observed variables but CFA involves both observed and latent variables
    - 2. EFA estimates all possible loadings but CFA usually only estimates some
    - 3. EFA can include multiple latent variables but CFA can only include one
    - 4. CFA tests causality while EFA can only test correlation

Welcome back 1

- Welcome back!
- The answer to the quiz question is…
    - *1. EFA only involves observed variables but CFA involves both observed and latent variables*
    - *2. **EFA estimates all possible loadings but CFA usually only estimates some***
    - *3. EFA can include multiple latent variables but CFA can only include one*
    - *4. CFA tests causality while EFA can only test correlation*

The CFA model



The parameters of a CFA

- Latent factor variances and covariances
    - *The variability of and associations between the latent factors*
- Factor loadings
    - *Regression of the latent variables on the observed variables*
    - *Strength of relation between underlying latent variables and observed variables*
- Residual variances
    - *Variance in the observed variables not explained by the latent variables*
- Residual covariances
    - *The covariances between observed variables that exist over and above their covariance due to their shared relation with a latent factor*
- CFAs involve finding (or specifying) values for all of these parameters

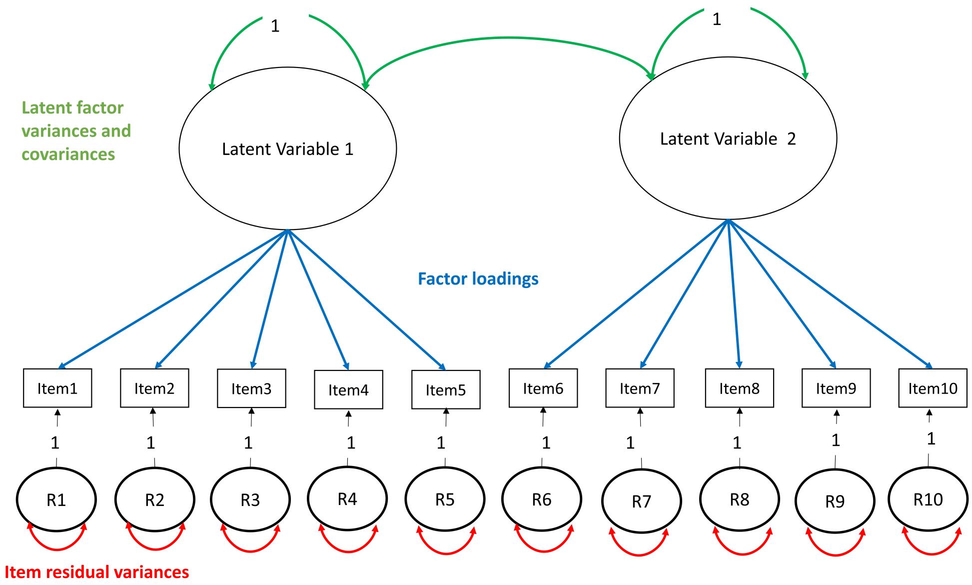
Model specification

- Defining the model we want to test
  - *i.e., which **parameters** do we want to estimate?*
    - How many factors?
    - Which items do we think go with which factors?
    - Are the factors correlated?
- Based on theory or past research (e.g., previous EFAs)

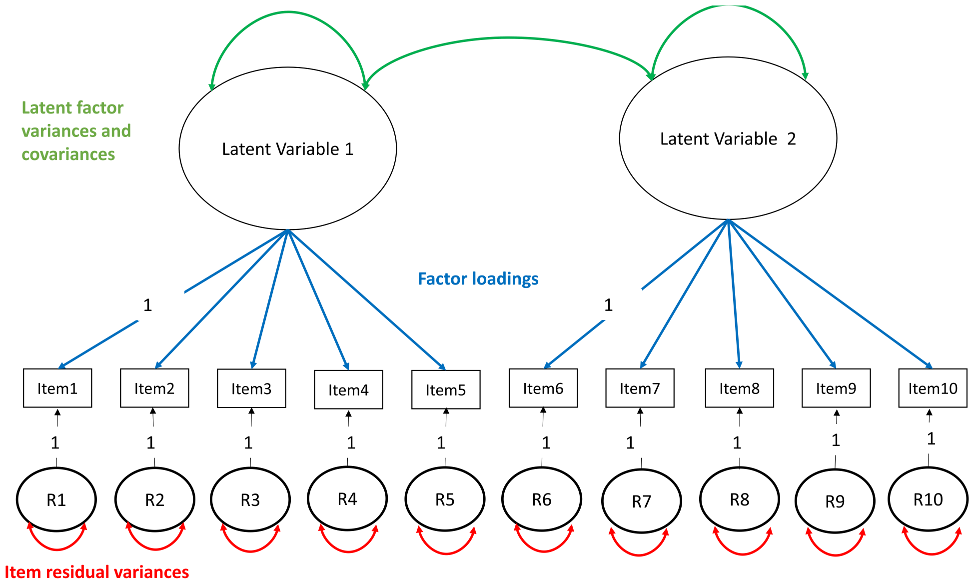
Latent variable scaling

- Latent variable scaling is a key aspect of model specification
- Latent variables have no inherent scale, so we have to define one
- Two commonly used scaling methods:
  - *Method 1: Fix the variance of each latent variable to 1*
  - *Method 2: Fix one factor loading for each latent variable to 1*
- Note that the necessity of scaling also applies to the residual factors
  - *Typically uses Method 2*

Scaling the latent variables by fixing factor variances



Scaling the latent variables by fixing factor loadings



Model identification

- More generally, we need to ensure that the model we specify is **identified**
- Identification concerns the number of ‘knowns’ versus ‘unknowns’
- There must be more knowns than unknowns in order to be able to test a CFA
- In CFA, the knowns are variances and covariances of the observed variables
- The unknowns are the parameters we want to estimate
- **Degrees of freedom** are the difference between knowns and unknowns

Levels of identification

- There are three levels of identification:
- **Under-identified** models: have < 0 degrees of freedom
- **Just Identified** models: have 0 degrees of freedom
- **Over-Identified** models: have > 0 degrees of freedom

Model identification illustration

- Chou & Bentler (1995) provide an illustration based on simultaneous linear equations:
  - *Eq.1:* $x + y = 5$
  - *Eq.2:* $2x + y = 8$
  - *Eq.3:* $x + 2y = 9$
- Eq.1 is on its own is *under-identified*
- Eq.1 & 2 are together *just identified*
- Eq.1, 2 & 3 are together *over identified*

The number of knowns

- To ensure model identification, we need to know the number of knowns
- We can calculate the knowns by:

$$\frac{(k+1)\,(k)}{2}$$

- where *k* is the number of observed variables.

The number of knowns

- This is the number of unique elements in the variance-covariance matrix for our observed variables
  - *e.g., if we had three observed variables:*

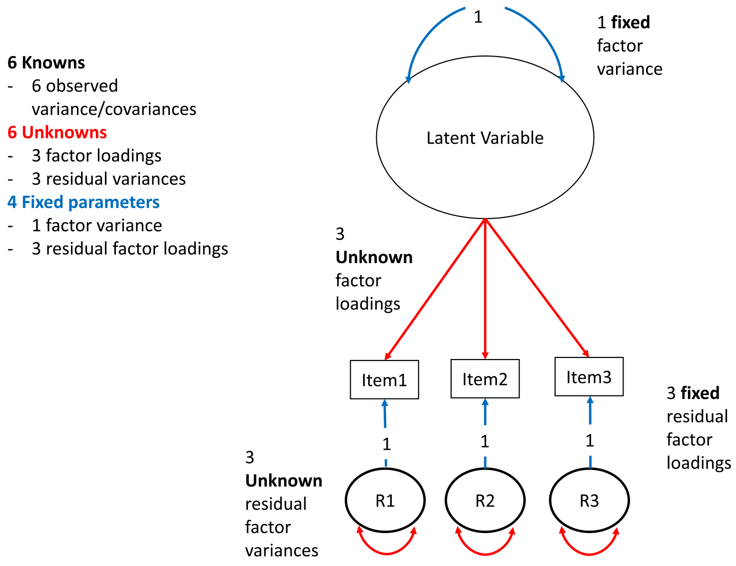
```
round(cov(Three.variables),2)
```

```
##      V1   V2   V3
## V1 1.05 0.33 0.42
## V2 0.33 1.03 0.65
## V3 0.42 0.65 1.01
```

- We have 6 unique elements (3 variances and 3 covariances)

Implications for CFA

- We usually need a minimum of three observed variables for a just identified model



BREAK 2

- Time for a pause
- Quiz question:
  - *A CFA model with 3 degrees of freedom would be best described as:*
    - Under-identified
    - Over-identified
    - Just identified
    - Negatively identified

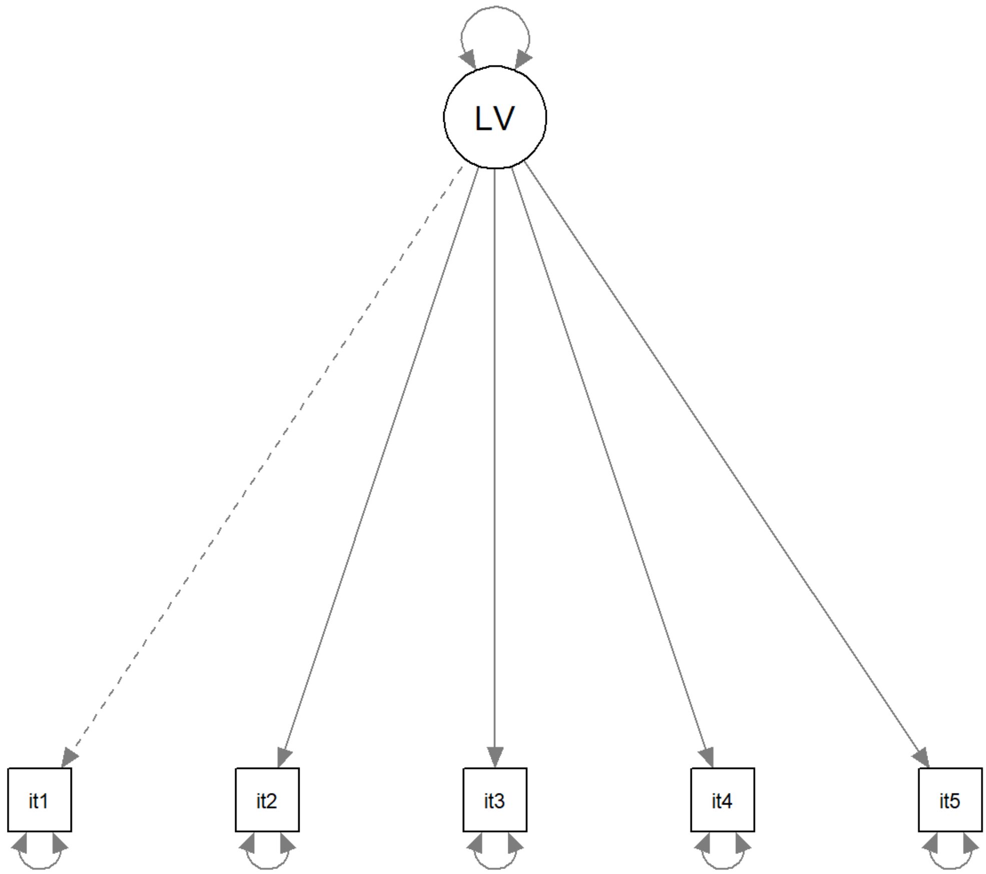
Welcome back 2

- Welcome back!
- The answer to the quiz question is:
  - *A CFA model with 3 degrees of freedom would be best described as:*
    - Under-identified
    - **Over-identified**
    - Just identified
    - Negatively identified

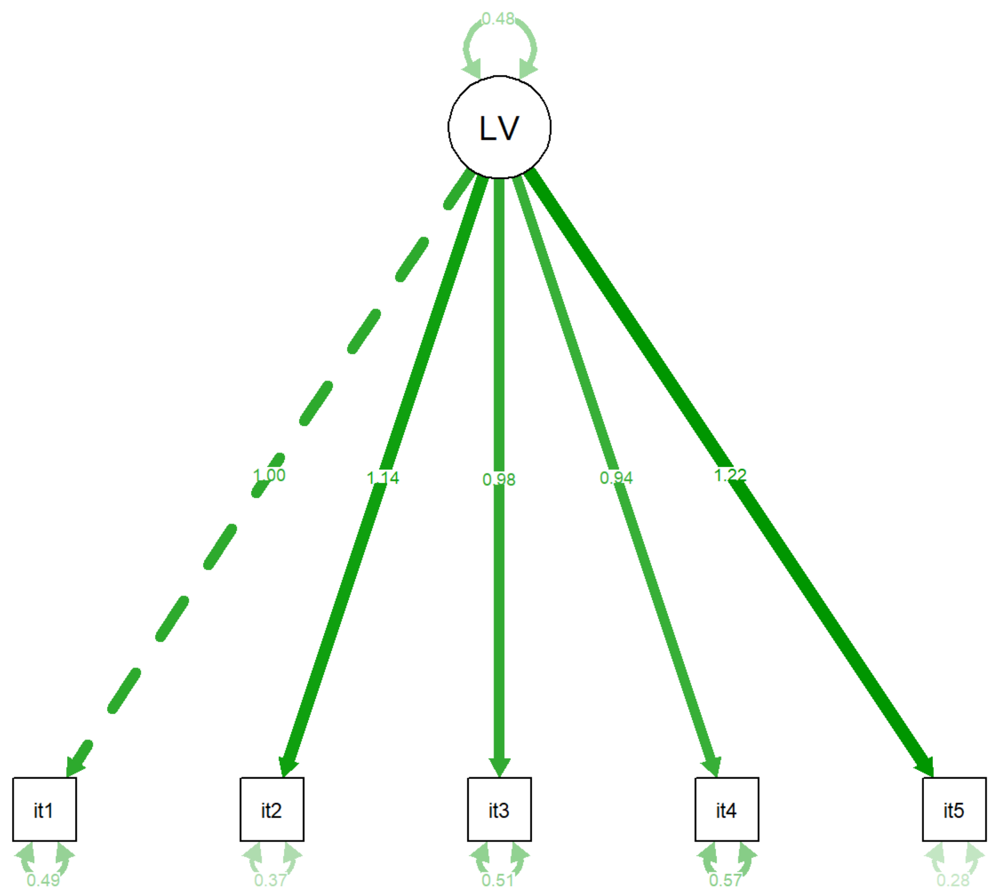
Model estimation

- After we have specified our model (& checked it is identified) we proceed to **estimation**
- Model estimation refers to finding the ‘best’ values for the unknown parameters

Specifying which parameters to estimate…



Finding the parameter values



Maximum likelihood estimation

- Maximum likelihood estimation is most commonly used
- Finds the parameters that maximise the likelihood of the data
- Begins with a set of starting values
- Iterative process of improving these values
  - *i.e. to minimise the difference between the sample covariance matrix and the covariance matrix implied by the parameter values*
- Terminates when the values are no longer substantially improved across iterations
  - *At this point **convergence** is said to have been reached*

No convergence?

- Sometimes estimation fails
- Common reasons are:
  - *The model is not identified*
  - *The model is very mis-specified*
  - *The model is very complex so more iterations are needed than the program default*

Maximum likelihood estimation assumptions

- Large sample size
- Multivariate normality
- Variables are on a continuous scale

Alternative estimators

- Robust maximum likelihood estimation
  - *For non-normal data*
- Weighted least squares, unweighted least squares or diagonally weighted least squares
  - *For ordinal data*

BREAK 3

- Time for a pause
- Quiz question:
  - *Which of these is most likely to result in a model failing to converge using maximum likelihood estimation:*
    - Model under-identification
    - Skewed variables
    - A sample size of only n=100
    - Ordinal variables with 5 response options

Welcome back 3

- Welcome back!
- The answer to the quiz question is…
    - *Which of these is most likely to result in a model failing to converge using maximum likelihood estimation:*
        - **Model under-identification**
        - Skewed variables
        - A sample size of only n=100
        - Ordinal variables with 5 response options

Model evaluation

- Once the model has been evaluated, we ask: *how good is the model?*
  - *Global fit*
  - *Local fit*

Global fit

- χ^2
  - *When we use maximum likelihood estimation we obtain a χ^2 value for the model*
  - *This can be compared to a χ^2 distribution with degrees of freedom equal to our model degrees of freedom*
  - *Statistically significant χ^2 suggests the CFA model does not do a good job of reproducing the observed variance-covariance matrix*
- However, χ^2 does not work well in practice
  - *Leads to the rejection of models that are only trivially mis-specified*

Alternatives to χ^2

- Absolute fit
  - *Standardised root mean square residual (**SRMR**)*
    - measures the discrepancy between the observed correlation matrix and model-implied correlation matrix
    - ranges from 0 to 1 with 0=perfect fit
    - values <.05 considered good
- Parsimony-corrected
  - *Corrects for the complexity of the model*
  - *Adds a penalty for having more degrees of freedom*
  - *Root mean square square error of approximation (**RMSEA**)*
    - 0=perfect fit
    - values <.05 considered good

Incremental fit indices

- Compares the model to a more restricted baseline model
  - *Usually an 'independence' model where all observed variable covariances fixed to 0*
- Comparative fit index (**CFI**)
  - *ranges between 0 and 1 with 1=perfect fit*
  - *values > .95 considered good*
- Tucker-Lewis index (**TLI**)
  - *includes a parsimony penalty*
  - *values >.95 considered good*

Local fit

- It is also possible to examine **local** areas of mis-fit
- **Modification indices** estimate the improvement in χ^2 that could be expected from including an additional parameter
    - *e.g., a cross-loading, residual covariance or latent variable covariance*
- **Expected parameter changes** estimate the value of the parameter were it to be included

Making model modifications

- Modification indices and expected parameter changes can be helpful for identifying how to improve the model
- However:
  - *Modifications should be made iteratively*
  - *Don't go overboard: may just be capitalising on chance*
  - *Make sure the modifications can be justified on substantive grounds*
  - *Be aware that this becomes an exploratory modeling practice*
  - *Ideally replicate the new model in an independent sample*

Other considerations in model evaluation

- Ideally:
  - *Factor loadings should be statistically significant*
  - *Standardised factor loadings should be >|.30|*
  - *Else some items/parameters could be trimmed from the model*
  - *(with the same caveats as on previous slide)*
- Check for **Heywood cases**
  - *Parameter estimates that are outside the permissable range*
  - *E.g., correlations >1, negative residual variances*
  - *May require modifications to the model to address*

BREAK 4

- Time for a pause
- Quiz question
- Which of these fit indices compares a CFA model fit to an ‘independence model’ fit:
    - *Comparative fit index (CFI)*
    - *Root Mean Square Error of Approximation (RMSEA)*
    - *Standardised Root Mean Square Residual (SRMR)*
    - *Chi-square*

Welcome back 4

- Welcome back!
- Which of these fit indices compares a CFA model fit to an ‘independence model’ fit:
    - ***Comparative fit index (CFI)***
    - *Root Mean Square Error of Approximation (RMSEA)*
    - *Standardised Root Mean Square Residual (SRMR)*
    - *Chi-square*

Interpreting a CFA

- To aid interpretation we can request a fully **standardised solution**
- Converts loadings/covariances to a correlation metric
- Thereafter, the interpretation is similar to EFA:
    - *Loadings tell us strength of association between latent factor and items*
    - *Factor correlations tell us how strongly associated latent factors are*

Conducting a CFA model using lavaan

- Lavaan = **La**tent **Va**riable **An**alysis
- Used to specify and estimate latent variable models
- Three steps:

```
#step 1: specify the model

model<-'LV=~V1+V2+V3+V4'
  # we write the model using lavaan syntax enclosed in single quote marks

#step2: estimate the model

model.est<-cfa(model=model, data=df)
   # 'model= ' refers to a lavaan syntax object with the model specification
   # 'data= ' gives name of the dataframe in which to find the variables
#step3: inspect the results

summary(model.est)
    # the summary function shows us output from a fitted model
```

Model specification

- Specification uses lavaan syntax:

```
# simple regression model

Regression<-'DV~IV'

# multiple regression model

Multiple.regression<-'DV~IV1+IV2+IV3'

#covariance between two variables

Covariance<-'V1~~V2'

#latent factor specification

CFA<-'LV=~V1+V2+V3+V4'
```

Model specification for our aggression example

1. I hit someone
2. I kicked someone
3. I shoved someone
4. I battered someone
5. I physically hurt someone on purpose
6. I deliberately insulted someone
7. I swore at someone
8. I threatened to hurt someone
9. I called someone a nasty name to their face
10. I shouted mean things at someone

```
agg_m<-
  'Pagg=~item1+item2+item3+item4+item5

   Vagg=~item6+item7+item8+item9+item10

   Pagg~~Vagg'
```

Model estimation in lavaan

- To estimate the model, we then feed the object we just created into the cfa( ) function
- We also name the dataset containing the model
  - *Lavaan will compute the variance-covariance matrix internally*

```
agg_m.est<-cfa(agg_m, data=agg.items)
```

Scaling constraints

- By default, cfa( ) will scale the latent variables by fixing the first indicator for each latent factor to 1
- To override this and fix latent factor variances instead, we can write:

```
agg_m.est<-cfa(agg_m, data=agg.items, std.lv=T)
```

Model evaluation

- We can check the model fit using the summary( ) function:

```
summary(agg_m.est, fit.measures=T)
```

```
## lavaan 0.6-5 ended normally after 23 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                          1000
##
## Model Test User Model:
##
##   Test statistic                                41.739
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.170
##
## Model Test Baseline Model:
##
##   Test statistic                              4711.354
##   Degrees of freedom                                45
##   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)             -11838.328
##   Loglikelihood unrestricted model (H1)     -11817.459
##
##   Akaike (AIC)                               23718.657
##   Bayesian (BIC)                             23821.720
##   Sample-size adjusted Bayesian (BIC)        23755.023
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.015
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.029
##   P-value RMSEA <= 0.05                          1.000
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.024
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Pagg =~
##     item1             0.695    0.029   24.157    0.000
##     item2             0.792    0.028   28.463    0.000
##     item3             0.678    0.029   23.387    0.000
##     item4             0.656    0.030   22.003    0.000
##     item5             0.850    0.027   31.231    0.000
##   Vagg =~
##     item6             0.652    0.030   21.967    0.000
##     item7             0.873    0.025   34.276    0.000
##     item8             0.922    0.025   36.289    0.000
##     item9             0.662    0.029   23.157    0.000
##     item10            0.691    0.029   24.006    0.000
```

```
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   Pagg ~~
##     Vagg              0.098    0.035    2.792    0.005
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .item1             0.491    0.026   19.051    0.000
##    .item2             0.369    0.022   16.488    0.000
##    .item3             0.515    0.027   19.360    0.000
##    .item4             0.572    0.029   19.844    0.000
##    .item5             0.286    0.021   13.748    0.000
##    .item6             0.605    0.029   20.891    0.000
##    .item7             0.219    0.016   13.978    0.000
##    .item8             0.168    0.015   10.988    0.000
##    .item9             0.541    0.026   20.657    0.000
##    .item10            0.532    0.026   20.469    0.000
##     Pagg              1.000
##     Vagg              1.000
```

Model evaluation

- We can examine local mis-specifications using the modindices() function

```
modindices(agg_m.est, sort=T)
```

```
##      lhs op    rhs    mi    epc sepc.lv sepc.all sepc.nox
## 26  Pagg =~  item8 7.281 -0.051  -0.051   -0.051   -0.051
## 27  Pagg =~  item9 7.099  0.069   0.069    0.070    0.070
## 25  Pagg =~  item7 6.654  0.050   0.050    0.050    0.050
## 38 item1 ~~  item6 5.283 -0.044  -0.044   -0.080   -0.080
## 61 item4 ~~  item8 4.760 -0.030  -0.030   -0.096   -0.096
## 24  Pagg =~  item6 3.288 -0.050  -0.050   -0.049   -0.049
## 57 item3 ~~ item10 2.572 -0.029  -0.029   -0.056   -0.056
## 60 item4 ~~  item7 2.231  0.021   0.021    0.059    0.059
## 33  Vagg =~  item5 2.018 -0.032  -0.032   -0.032   -0.032
## 36 item1 ~~  item4 1.997 -0.028  -0.028   -0.054   -0.054
## 53 item3 ~~  item6 1.701  0.025   0.025    0.045    0.045
## 40 item1 ~~  item8 1.513  0.016   0.016    0.055    0.055
## 62 item4 ~~  item9 1.427  0.023   0.023    0.041    0.041
## 46 item2 ~~  item6 1.333 -0.020  -0.020   -0.043   -0.043
## 56 item3 ~~  item9 1.290  0.021   0.021    0.040    0.040
## 69 item6 ~~  item7 1.050  0.017   0.017    0.046    0.046
## 59 item4 ~~  item6 0.929  0.019   0.019    0.033    0.033
## 49 item2 ~~  item9 0.816  0.015   0.015    0.033    0.033
## 47 item2 ~~  item7 0.815  0.011   0.011    0.038    0.038
## 70 item6 ~~  item8 0.760 -0.015  -0.015   -0.046   -0.046
## 55 item3 ~~  item8 0.730 -0.011  -0.011   -0.038   -0.038
## 34 item1 ~~  item2 0.698  0.016   0.016    0.038    0.038
## 77 item8 ~~ item10 0.678  0.014   0.014    0.046    0.046
## 32  Vagg =~  item4 0.642  0.021   0.021    0.021    0.021
## 51 item3 ~~  item4 0.615  0.016   0.016    0.029    0.029
## 67 item5 ~~  item9 0.611 -0.012  -0.012   -0.031   -0.031
## 75 item7 ~~ item10 0.538 -0.012  -0.012   -0.035   -0.035
## 48 item2 ~~  item8 0.524 -0.008  -0.008   -0.034   -0.034
## 54 item3 ~~  item7 0.503  0.009   0.009    0.028    0.028
## 45 item2 ~~  item5 0.294 -0.012  -0.012   -0.036   -0.036
## 76 item8 ~~  item9 0.275  0.009   0.009    0.028    0.028
## 41 item1 ~~  item9 0.262  0.009   0.009    0.018    0.018
## 43 item2 ~~  item3 0.246 -0.010  -0.010   -0.022   -0.022
## 30  Vagg =~  item2 0.229  0.011   0.011    0.011    0.011
## 50 item2 ~~ item10 0.214  0.008   0.008    0.017    0.017
## 37 item1 ~~  item5 0.182  0.008   0.008    0.023    0.023
## 29  Vagg =~  item1 0.164  0.010   0.010    0.010    0.010
## 28  Pagg =~ item10 0.161 -0.010  -0.010   -0.010   -0.010
## 74 item7 ~~  item9 0.133 -0.006  -0.006   -0.017   -0.017
## 63 item4 ~~ item10 0.123  0.007   0.007    0.012    0.012
## 78 item9 ~~ item10 0.107 -0.006  -0.006   -0.011   -0.011
## 65 item5 ~~  item7 0.094 -0.004  -0.004   -0.014   -0.014
## 58 item4 ~~  item5 0.076  0.005   0.005    0.013    0.013
## 31  Vagg =~  item3 0.065  0.007   0.007    0.007    0.007
## 35 item1 ~~  item3 0.055 -0.005  -0.005   -0.009   -0.009
## 71 item6 ~~  item9 0.053 -0.005  -0.005   -0.008   -0.008
## 68 item5 ~~ item10 0.051  0.004   0.004    0.009    0.009
## 72 item6 ~~ item10 0.048  0.004   0.004    0.008    0.008
## 44 item2 ~~  item4 0.038  0.004   0.004    0.008    0.008
## 73 item7 ~~  item8 0.034 -0.004  -0.004   -0.022   -0.022
## 64 item5 ~~  item6 0.021 -0.002  -0.002   -0.006   -0.006
## 66 item5 ~~  item8 0.020 -0.002  -0.002   -0.007   -0.007
## 39 item1 ~~  item7 0.020 -0.002  -0.002   -0.006   -0.006
## 42 item1 ~~ item10 0.005 -0.001  -0.001   -0.003   -0.003
## 52 item3 ~~  item5 0.003  0.001   0.001    0.003    0.003
```

Standardised parameter estimates

- We can also inspect the standardised parameter estimates

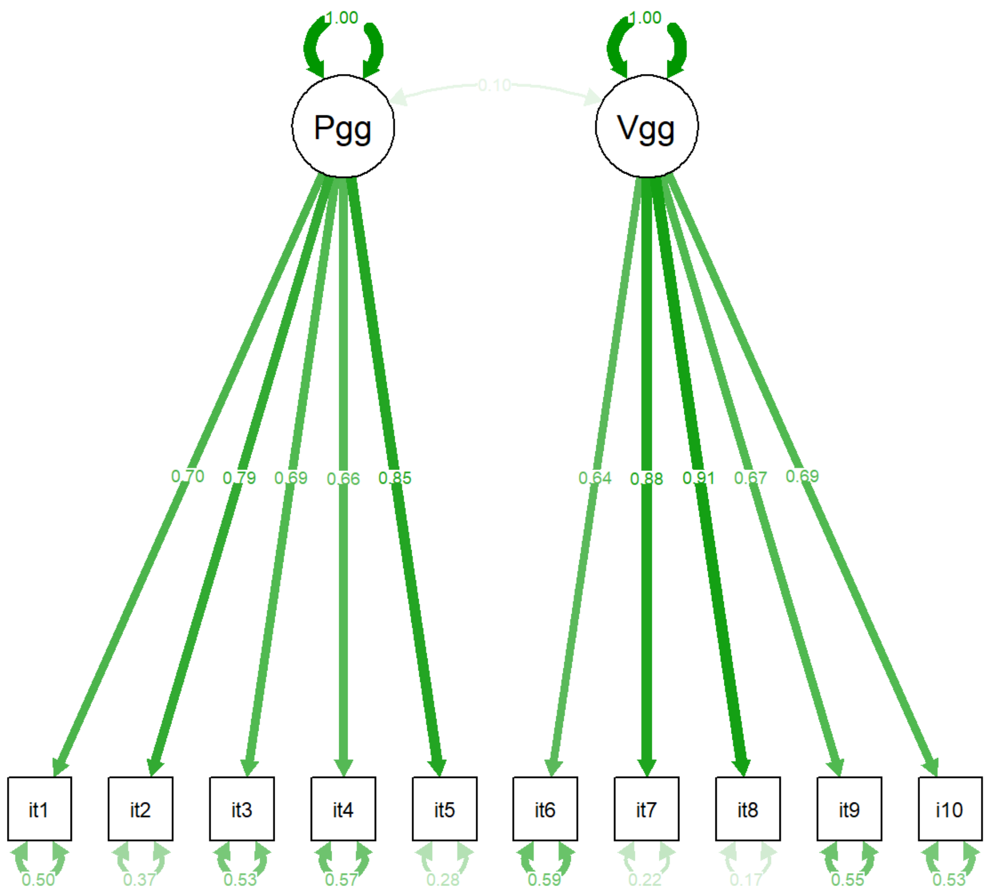
```
summary(agg_m.est, standardized=T)
```

```
## lavaan 0.6-5 ended normally after 23 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                          1000
##
## Model Test User Model:
##
##   Test statistic                                41.739
##   Degrees of freedom                                34
##   P-value (Chi-square)                           0.170
##
## Parameter Estimates:
##
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##   Standard errors                             Standard
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pagg =~
##     item1             0.695    0.029   24.157    0.000    0.695    0.704
##     item2             0.792    0.028   28.463    0.000    0.792    0.793
##     item3             0.678    0.029   23.387    0.000    0.678    0.687
##     item4             0.656    0.030   22.003    0.000    0.656    0.655
##     item5             0.850    0.027   31.231    0.000    0.850    0.846
##   Vagg =~
##     item6             0.652    0.030   21.967    0.000    0.652    0.642
##     item7             0.873    0.025   34.276    0.000    0.873    0.881
##     item8             0.922    0.025   36.289    0.000    0.922    0.914
##     item9             0.662    0.029   23.157    0.000    0.662    0.669
##     item10            0.691    0.029   24.006    0.000    0.691    0.688
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##   Pagg ~~
##     Vagg              0.098    0.035    2.792    0.005    0.098    0.098
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
##    .item1             0.491    0.026   19.051    0.000    0.491    0.504
##    .item2             0.369    0.022   16.488    0.000    0.369    0.371
##    .item3             0.515    0.027   19.360    0.000    0.515    0.528
##    .item4             0.572    0.029   19.844    0.000    0.572    0.570
##    .item5             0.286    0.021   13.748    0.000    0.286    0.284
##    .item6             0.605    0.029   20.891    0.000    0.605    0.588
##    .item7             0.219    0.016   13.978    0.000    0.219    0.223
##    .item8             0.168    0.015   10.988    0.000    0.168    0.165
##    .item9             0.541    0.026   20.657    0.000    0.541    0.552
##    .item10            0.532    0.026   20.469    0.000    0.532    0.527
##     Pagg              1.000                               1.000    1.000
##     Vagg              1.000                               1.000    1.000
```

Visualising the model

- Sempaths() from the semPlot package can be used to visual a model as a SEM diagram

```
semPaths(agg_m.est, what='stand')
```



Writing up a CFA model

- Methods
    - *Model(s) being tested*
    - *Scaling /identifcation method*
    - *Estimation method*
    - *Criteria that used to judge fit*
- Results
    - *Model fit (χ^2 test, CFI, TLI, RMSEA, SRMR)*
    - *Any modifications made and why*
    - *Model parameters (in a SEM diagram or table)*

Cautions regarding CFA

- Good fit doesn't guarantee that the model is 'correct'
- Be careful about 'reifying' latent variables
- Even when there are no common factors, CFA models can fit well

Summary

CFA involves testing a hypothesised factor structure

- Specifying a model
  - *Identification and scaling*
- Estimating that model
  - *e.g., maximum likelihood estimation*
- Seeing how well that model fits the data
  - *Global and local fit*
- Interpreting the model