Example 4: CFA of Forgiveness of Situations ($N = 1103$) in M*plus* v. 8.8, R lavaan 0.6-12, and a little bit of SAS MIXED (complete syntax and output available electronically)

This example comes from the Heartland Forgiveness Scale (Yamhure Thompson et al., 2005). Here we focus on the **Forgiveness of Situations** subscale with six items, three of which are reverse-coded, each rated on a 7-point scale:

1. *When things go wrong for reasons that can't be controlled, I get stuck in negative thoughts about it.* ***(R)***
2. *With time I can be understanding of bad circumstances in my life.*
3. *If I am disappointed by uncontrollable circumstances in my life, I continue to think negatively about them.* ***(R)***
4. *I eventually make peace with bad situations in my life.*
5. *It's really hard for me to accept negative situations that aren't anybody's fault.* ***(R)***
6. *Eventually I let go of negative thoughts about bad circumstances that are beyond anyone's control.*

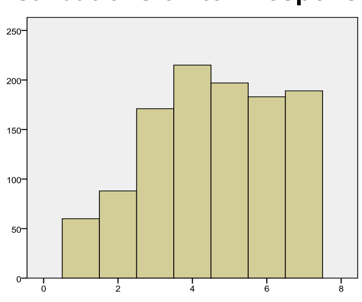
Response Anchors: 1 = Almost Always False of Me, 2=?, 3 = More Often False of Me, 4 = ?,
5 = More Often True of Me, 6 = ?, 7 = Almost Always True of Me

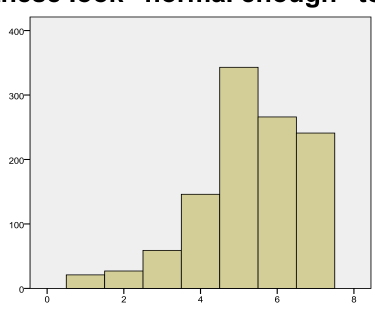
Observed Correlation Matrix	**R1**	**2**	**R3**	**4**	**R5**	**6**
R1	**1.000**					
2	0.240	**1.000**				
R3	0.647	0.317	**1.000**			
4	0.300	0.570	0.369	**1.000**		
R5	0.453	0.255	0.482	0.289	**1.000**	
6	0.297	0.457	0.356	0.448	0.304	**1.000**
Means	4.547	5.289	4.896	5.359	4.860	5.321
Variances	3.049	1.903	2.543	1.967	2.945	2.341

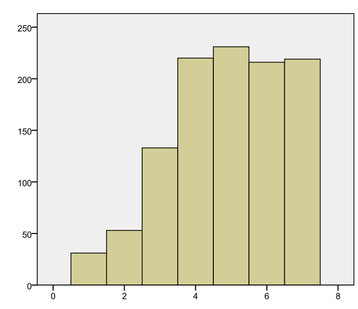
Observed Covariance Matrix	**R1**	**2**	**R3**	**4**	**R5**	**6**
R1	**3.049**					
2	0.577	**1.903**				
R3	1.802	0.697	**2.543**			
4	0.734	1.103	0.824	**1.967**		
R5	1.358	0.604	1.319	0.695	**2.945**	
6	0.795	0.965	0.868	0.962	0.798	**2.341**

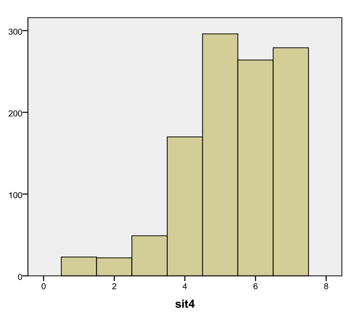
To do CFA modeling, you only really need means, variances, and either correlations or covariances among items:
$\boldsymbol{Cov}(y_1, y_2) = \boldsymbol{Cor}(y_1, y_2) * \boldsymbol{SD}(y_1) * \boldsymbol{SD}(y_2)$ OR $\boldsymbol{Cor}(y_1, y_2) = \boldsymbol{Cov}(y_1, y_2) \,/\, \boldsymbol{SD}(y_1) * \boldsymbol{SD}(y_2)$

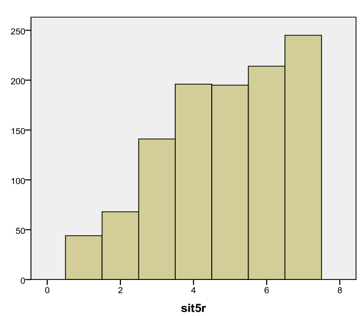
Distributions of item responses – do these look "normal enough" to you?

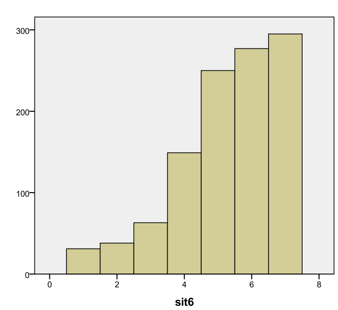












Mplus Code to Read in Data and Select Options Across Models
(Note: DO NOT copy syntax from this handout—start from the original .inp file instead):

```
TITLE:        CFA of Situation Factor
DATA:         FILE = Example4.csv;       ! Don't need path if in same directory
              FORMAT = free;             ! Default
              TYPE = INDIVIDUAL;         ! Default

VARIABLE:     NAMES = PersonID Self1 Self2r Self3 Self4r Self5 Self6r
                      Other1r Other2 Other3r Other4 Other5r Other6
                      Sit1r Sit2 Sit3r Sit4 Sit5r Sit6
                      Selfsub Othsub Sitsub HFSsum;                 ! Every variable in DATASET

              USEVARIABLES = Sit1r Sit2 Sit3r Sit4 Sit5r Sit6;      ! Every variable in MODEL
              MISSING = ALL (99999);                                ! Identify missing values
              IDVARIABLE = PersonID;                                ! Identify person ID variable

ANALYSIS:     TYPE = GENERAL;     ! Default
              ESTIMATOR = MLR;    ! Robust ML

OUTPUT:       MODINDICES (6.635); ! Cheat codes to improve the model at p <.01 for df=1
              STDYX;              ! Fully standardized solution
              RESIDUAL;           ! Standardized and normalized residuals for local fit

!SAVEDATA:    SAVE = FSCORES; FILE = FactorScores.dat;  ! To save factor scores (optional)

!PLOT:        TYPE = PLOT1 PLOT2 PLOT3; ! To get plots (factor score distributions) as needed

MODEL:      ! (model syntax goes here, to be changed for each model as shown below)
```

Mplus Syntax for Model 1: Single Factor Using Fully Z-Scored Factor Scaling
(Factor Variance = 1, Factor Mean = 0, So All Item Loadings and Intercepts Estimated)

The following code refers to EVERY model parameter for completeness:

```
! Model 1: Single Factor Using Fully Z-Scored Factor Scaling

    ! Item factor loadings --> @=fixed, *=free → * REQUIRED for first item if free
      Sit BY Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;

    ! Item intercepts --> [ ] indicates means or intercepts, @=fixed, *=free
      [Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*];

    ! Item error variances --> just list item by itself, @=fixed, *=free
      Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;

    ! Factor variance fixed to 1 via @1 --> just list factor by itself, @=fixed, *=free
      Sit@1;

    ! Factor mean fixed to 0 via @0 --> [ ] for means or intercepts, @=fixed, *=free
      [Sit@0];
```

In reality, all you'd need to write to define this model is this:

```
    ! Item factor loadings --> @=fixed, *=free → * REQUIRED for first item if free
      Sit BY Sit1r* Sit2 Sit3r Sit4 Sit5r Sit6;

    ! Factor variance --> just list factor by itself, @=fixed, *=free
      Sit@1;
```

By default, all intercepts are estimated separately and the factor mean is fixed at 0.
By default, all residual variances for the items are estimated separately, too.
By default, factor variances and covariances are estimated freely.

Mplus Output for Model 1: Single Factor Using Fully Z-Scored Factor Scaling
(Factor Variance = 1, Factor Mean = 0, So All Item Loadings and Intercepts Estimated)

```
UNSTANDARDIZED MODEL RESULTS
                                                    Two-Tailed
                    Estimate       S.E.  Est./S.E.    P-Value

FACTOR LOADINGS (regression slopes of item response on factor)
SIT      BY
    SIT1R              1.234      0.069     17.906      0.000
    SIT2               0.702      0.074      9.441      0.000
    SIT3R              1.241      0.063     19.846      0.000
    SIT4               0.784      0.069     11.334      0.000
    SIT5R              1.023      0.053     19.179      0.000
    SIT6               0.819      0.069     11.942      0.000

Means (of Factor)
999 = "cannot be computed" – you will see 999 for any parameter that is FIXED
    SIT                0.000      0.000    999.000    999.000

Intercepts (of Items) – HERE, ARE ACTUAL ITEM MEANS BECAUSE FACTOR MEAN IS ZERO
    SIT1R              4.547      0.053     86.474      0.000
    SIT2               5.289      0.042    127.347      0.000
    SIT3R              4.896      0.048    101.959      0.000
    SIT4               5.359      0.042    126.895      0.000
    SIT5R              4.860      0.052     94.060      0.000
    SIT6               5.321      0.046    115.493      0.000

Variances (of Factor)
999 = "cannot be computed" – here, because the parameter is fixed to 1 already
    SIT                1.000      0.000    999.000    999.000

 Residual Variances (variance of e's)
    SIT1R              1.526      0.149     10.217      0.000
    SIT2               1.409      0.128     11.014      0.000
    SIT3R              1.004      0.135      7.456      0.000
    SIT4               1.352      0.127     10.672      0.000
    SIT5R              1.899      0.118     16.025      0.000
    SIT6               1.671      0.159     10.517      0.000
```

Making use of the unstandardized model estimates:

<u>Writing out the model—individual predicted values:</u>

$y_{1s} = \mu_1 + \lambda_1 F_s + e_{1s}$

$y_{1s} = 4.547 + 1.234 F_s + e_{1s}$

Stay tuned! **1.523 will become the factor variance when item 1 is used as the "marker" item** (whose loading is fixed to 1): 1.523 is the amount of item 1's variance that is due to the factor (with 1.526 error due to "not the factor"

<u>Writing out the model—predicted item variances and covariances:</u>

$Var(y_1) = (\lambda_1^2)\, Var(F) + Var(e_1)$

$Var(y_1) = (1.234^2)(1) + 1.526 = \mathbf{1.523} + 1.526 = 3.049$ (= original item variance)

$Cov(y_1, y_2) = \lambda_1 * Var(F) * \lambda_2$

$Cov(y_1, y_2) = 1.234 * 1 * 0.702 = 0.866$

(actual covariance = 0.577, so the model over-predicted how related items 1 and 2 should be)

MPLUS STDYX STANDARDIZED MODEL RESULTS (FULLY STANDARDIZED WITH RESPECT TO X & Y)

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value
FACTOR LOADINGS (correlations of item response with factor)				
Square these to get reliability (proportion "true variance") per item				
SIT BY				
SIT1R	0.707	0.035	19.983	0.000
SIT2	0.509	0.053	9.545	0.000
SIT3R	0.778	0.034	22.655	0.000
SIT4	0.559	0.048	11.641	0.000
SIT5R	0.596	0.029	20.528	0.000
SIT6	0.535	0.047	11.392	0.000
Means (of Factor)				
SIT	0.000	0.000	999.000	999.000
Intercepts (of Items) → is intercept / SD(Y) → not usually reported				
SIT1R	2.604	0.057	45.888	0.000
SIT2	3.834	0.111	34.394	0.000
SIT3R	3.070	0.072	42.921	0.000
SIT4	3.821	0.111	34.441	0.000
SIT5R	2.832	0.066	43.095	0.000
SIT6	3.477	0.101	34.573	0.000
Variances (of Factor) → will always be 1 in a standardized solution				
SIT	1.000	0.000	999.000	999.000
Residual Variances (standardized variance of e residuals)				
SIT1R	0.500	0.050	10.009	0.000
SIT2	0.741	0.054	13.628	0.000
SIT3R	0.395	0.053	7.388	0.000
SIT4	0.687	0.054	12.786	0.000
SIT5R	0.645	0.035	18.619	0.000
SIT6	0.714	0.050	14.187	0.000
R-SQUARE (equals 1-residual variance OR standardized loading squared)				
SIT1R	0.500	0.050	9.991	0.000
SIT2	0.259	0.054	4.772	0.000
SIT3R	0.605	0.053	11.327	0.000
SIT4	0.313	0.054	5.821	0.000
SIT5R	0.355	0.035	10.264	0.000
SIT6	0.286	0.050	5.696	0.000

The standardized solution will look identical across methods of model identification with respect to the factor loadings, error variances, and R-square values for the items. The standardized intercepts will change because they depend on the unstandardized intercepts (but the standardized intercepts are rarely reported or interpreted anyway).

Making use of the standardized model estimates:

<u>Writing out the model – predicted item correlations:</u>

$$Cor(y_1, y_2) = \lambda_1 * Var(F) * \lambda_2$$

$$Cor(y_1, y_2) = .707 * 1 * .509 = .360$$

(actual correlation = $.240$, so the model over-predicted how related items 1 and 2 should be)

R Syntax for Model 1: Single Factor Using Fully Z-Scored Factor Scaling
(Factor Variance = 1, Factor Mean = 0, So All Item Loadings and Intercepts Estimated)

The following code refers to EVERY model parameter for completeness:

```
# R Syntax for lavaan function: longer but more transparent version of model
Syntax1 = "
# Define factor and request item factor loadings --> factor =~ item + item + item
  Sit =~ Sit1r + Sit2 + Sit3r + Sit4 + Sit5r + Sit6
# Item intercepts --> ~ 1 indicates means or intercepts
  Sit1r ~ 1; Sit2 ~ 1; Sit3r ~ 1; Sit4 ~ 1; Sit5r ~ 1; Sit6 ~ 1
# Item error variances or covariances --> use the ~~ command
  Sit1r ~~ Sit1r; Sit2 ~~ Sit2; Sit3r ~~ Sit3r; Sit4 ~~ Sit4; Sit5r ~~ Sit5r; Sit6 ~~ Sit6
# Factor variance fixed=1 and factor mean fixed=0 (* means fixed in lavaan, seriously)
  Sit ~~ 1*Sit; Sit ~ 0
"
# Use MLR estimation like in Mplus, z-scored latent variables (mean=0, SD=1)
  Model1 = lavaan(model=Syntax1, data=Example4, estimator="MLR", mimic="mplus", std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution
  summary(object=Model1, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

In reality, all you'd need to write to define this model is this:

```
# R Syntax for sem or cfa functions: shorter but less transparent version of model
  Syntax1short = "Sit =~ Sit1r + Sit2 + Sit3r + Sit4 + Sit5r + Sit6"
# Use MLR estimation like in Mplus, z-score latent factor (mean=0, SD=1)
  Model1short = sem(model=Syntax1short, data=Example4, estimator="MLR", mimic="mplus",
                    std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution (will be same as from lavaan)
  summary(object=Model1short, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

R Lavaan Output for Model 1: Single Factor Using Fully Z-Scored Factor Scaling
(Factor Variance = 1, Factor Mean = 0, So All Item Loadings and Intercepts Estimated)

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Sit =~
    Sit1r             1.234    0.069   17.906    0.000    1.234    0.707
    Sit2              0.702    0.074    9.441    0.000    0.702    0.509
    Sit3r             1.241    0.063   19.847    0.000    1.241    0.778
    Sit4              0.784    0.069   11.333    0.000    0.784    0.559
    Sit5r             1.023    0.053   19.179    0.000    1.023    0.596
    Sit6              0.819    0.069   11.942    0.000    0.819    0.535

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit1r             4.547    0.053   86.474    0.000    4.547    2.604
   .Sit2              5.289    0.042  127.346    0.000    5.289    3.834
   .Sit3r             4.896    0.048  101.959    0.000    4.896    3.070
   .Sit4              5.359    0.042  126.896    0.000    5.359    3.821
   .Sit5r             4.860    0.052   94.060    0.000    4.860    2.832
   .Sit6              5.321    0.046  115.492    0.000    5.321    3.477
    Sit               0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit1r             1.526    0.149   10.217    0.000    1.526    0.500
   .Sit2              1.409    0.128   11.014    0.000    1.409    0.741
   .Sit3r             1.004    0.135    7.456    0.000    1.004    0.395
   .Sit4              1.352    0.127   10.672    0.000    1.352    0.687
   .Sit5r             1.899    0.118   16.025    0.000    1.899    0.645
   .Sit6              1.671    0.159   10.517    0.000    1.671    0.714
    Sit               1.000                               1.000    1.000
```

For fully standardized results, use std.all = STDXY

Next up: two equivalent ways of getting the same model, but with different scaling
(i.e., <u>illustrating</u> the results of different methods of identification, even though model fit is the same)

Model 2. Single Factor Using Marker Item Loading = 1 and Factor Mean = 0
(Factor variance estimated given marker item loading=1, all intercepts estimated)

```
! Mplus Model 2: Single Factor Scaled Using Marker Item Loading=1 and Factor Mean=0
! Factor variance estimated given marker item loading=1, all intercepts estimated
    Sit BY Sit1r@1 Sit2* Sit3r* Sit4* Sit5r* Sit6*;      ! Loadings (1st fixed=1)
    [Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*];            ! Intercepts (all free)
     Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;             ! Residual variances (all free)
     Sit*;                                               ! Factor variance (free)
    [Sit@0];                                             ! Factor mean (fixed=0)
```

```
# R Syntax for lavaan function: longer but more transparent version of model
  Syntax2 = "
# Define factor and request item factor loadings with first loading fixed to 1
  Sit =~ 1*Sit1r + Sit2 + Sit3r + Sit4 + Sit5r + Sit6
# Item intercepts all estimated
  Sit1r ~ 1; Sit2 ~ 1; Sit3r ~ 1; Sit4 ~ 1; Sit5r ~ 1; Sit6 ~ 1
# Item error variances all estimated
  Sit1r ~~ Sit1r; Sit2 ~~ Sit2; Sit3r ~~ Sit3r; Sit4 ~~ Sit4; Sit5r ~~ Sit5r; Sit6 ~~ Sit6
# Factor variance=estimated and factor mean=0
  Sit ~~ Sit; Sit ~ 0
"
# Use MLR estimation like in Mplus, do not z-score latent factor (because variance is estimated)
  Model2 = lavaan(model=Syntax2, data=Example4, estimator="MLR", mimic="mplus", std.lv=FALSE)
# Print solution: get fit, get effect size, STDYX solution
  summary(object=Model2, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

# R Syntax for sem or cfa functions: shorter but less transparent version of model
  Syntax2short = "Sit =~ 1*Sit1r + Sit2 + Sit3r + Sit4 + Sit5r + Sit6"
  Model2short = sem(model=Syntax2short, data=Example4, estimator="MLR", mimic="mplus",
                    std.lv=FALSE)
```

MPLUS UNSTANDARDIZED MODEL RESULTS

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value
FACTOR LOADINGS (regression slopes of item response on factor)				
Here, loading for SIT1R is not tested because it is fixed=1				
SIT BY				
SIT1R	**1.000**	**0.000**	**999.000**	**999.000**
SIT2	0.569	0.083	6.830	0.000
SIT3R	1.005	0.035	28.555	0.000
SIT4	0.636	0.082	7.741	0.000
SIT5R	0.829	0.053	15.698	0.000
SIT6	0.664	0.081	8.143	0.000
Means (of Factor)				
SIT	0.000	0.000	999.000	999.000
Intercepts (of Items) – EXPECTED Y WHEN FACTOR = 0, or for mean of factor in sample				
SIT1R	**4.547**	0.053	86.474	0.000
SIT2	5.289	0.042	127.347	0.000
SIT3R	4.896	0.048	101.960	0.000
SIT4	5.359	0.042	126.896	0.000
SIT5R	4.860	0.052	94.060	0.000
SIT6	5.321	0.046	115.492	0.000
Variances (of Factor)				
SIT	**1.523**	**0.170**	**8.954**	**0.000**
Residual Variances (variances of e's)				
SIT1R	1.526	0.149	10.217	0.000
SIT2	1.409	0.128	11.014	0.000
SIT3R	1.004	0.135	7.456	0.000
SIT4	1.352	0.127	10.673	0.000
SIT5R	1.899	0.118	16.026	0.000
SIT6	1.671	0.159	10.517	0.000

PREVIOUSLY:

SIT	BY
SIT1R	1.234
SIT2	0.702
SIT3R	1.241
SIT4	0.784
SIT5R	1.023
SIT6	0.819

In Mplus, fixed parameters have SE=0 and test statistics = 999 (so 999 is an undefined value)

Stay tuned! **4.547 will become the factor mean** when item 1 is used as the "marker" item for the intercept (fixed to 0)

And here it is! **1.523** is the amount of item 1's variance that is due to the factor, and so it becomes the factor variance when item 1 is used as the "marker" item (loading fixed to 1 for identification).

R UNSTANDARDIZED MODEL RESULTS (TRUNCATED)

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  Sit =~
    Sit1r             1.000                               1.234    0.707
    Sit2              0.569    0.083    6.831    0.000    0.702    0.509
    Sit3r             1.005    0.035   28.553    0.000    1.241    0.778
    Sit4              0.636    0.082    7.741    0.000    0.784    0.559
    Sit5r             0.829    0.053   15.698    0.000    1.023    0.596
    Sit6              0.664    0.081    8.143    0.000    0.819    0.535
```

In R, fixed parameters have blanks for their SE and test statistics (instead of 999 as in Mplus)

Yet another equivalent alternative method for scaling the factor…

Model 3. Marker Item Loading = 1 and Intercept = 0 (Factor Variance and Mean Estimated)

```
! Mplus Model 3: Single Factor Scaled Using Marker Item Loading=1 and Intercept=0
! Factor variance and mean estimated given marker item loading=1 and intercept=0
    Sit BY Sit1r@1 Sit2* Sit3r* Sit4* Sit5r* Sit6*;      ! Loadings (1st fixed=1)
    [Sit1r@0 Sit2* Sit3r* Sit4* Sit5r* Sit6*];           ! Intercepts (1st fixed=0)
     Sit1r*  Sit2* Sit3r* Sit4* Sit5r* Sit6*;            ! Residual variances (all free)
     Sit*;                                               ! Factor variance (free)
    [Sit*];                                              ! Factor mean (free)
```

```
# R Syntax for lavaan function: longer but more transparent version of model
  Syntax3 = "
# Define factor and request item factor loadings with first loading fixed to 1
  Sit =~ 1*Sit1r + Sit2 + Sit3r + Sit4 + Sit5r + Sit6
# Item intercepts: first is fixed to 0 and rest estimated
  Sit1r ~ 0; Sit2 ~ 1; Sit3r ~ 1; Sit4 ~ 1; Sit5r ~ 1; Sit6 ~ 1
# Item error variances all estimated
  Sit1r ~~ Sit1r; Sit2 ~~ Sit2; Sit3r ~~ Sit3r; Sit4 ~~ Sit4; Sit5r ~~ Sit5r; Sit6 ~~ Sit6
# Factor variance=estimated and factor mean=estimated
  Sit ~~ Sit; Sit ~ 1
"
# Use MLR estimation like in Mplus, do not z-score latent factor (mean and variance estimated)
  Model3 = lavaan(model=Syntax3, data=Example4, estimator="MLR", mimic="mplus", std.lv=FALSE)
# Print solution: get fit, get effect size, STDYX solution
  summary(object=Model3, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

# No short version possible using sem that I know of
```

MPLUS UNSTANDARDIZED MODEL RESULTS (TRUNCATED)

```
Means (of Factor) → Note is mean of marker item 1
    SIT                  4.547      0.053     86.474      0.000

Intercepts (of Items) – EXPECTED Y WHEN FACTOR = 0
HERE, WHICH IS WHEN ITEM 1 = 0 → beyond scale of item, so values are very low
    SIT1R                0.000      0.000    999.000    999.000
    SIT2                 2.701      0.383      7.046      0.000
    SIT3R                0.325      0.171      1.899      0.058
    SIT4                 2.469      0.380      6.504      0.000
    SIT5R                1.092      0.246      4.431      0.000
    SIT6                 2.304      0.369      6.250      0.000
```

And here it is! **4.547** is the mean of item 1, which is now the factor mean when item 1's intercept is fixed to 0

R UNSTANDARDIZED MODEL RESULTS (TRUNCATED)

```
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit1r             0.000                               0.000    0.000
   .Sit2              2.701    0.383    7.045    0.000    2.701    1.958
   .Sit3r             0.325    0.171    1.899    0.058    0.325    0.204
   .Sit4              2.469    0.380    6.502    0.000    2.469    1.760
   .Sit5r             1.092    0.246    4.431    0.000    1.092    0.636
   .Sit6              2.304    0.369    6.249    0.000    2.304    1.506
    Sit               4.547    0.053   86.474    0.000    3.684    3.684
```

Calculating model degrees of freedom:

Total $\text{DF} = \frac{v(v+1)}{2} + v = \frac{6(6+1)}{2} + 6 = 27$

Spent: $\text{DF} = 18$

Leftover: Model $\text{DF} = 9$

Mplus model fit information for a single-factor model (same regardless of factor scaling method):

```
Number of Free Parameters                 18  → is # of estimated parameters ("free" to be not 0)

Loglikelihood – use for testing differences in model fit across nested models

  H0 Value                          -11536.404  → LL for YOUR specified model (bigger is better)
  H0 Scaling Correction Factor          1.4159  → indicates how far off from 1=normal
      for MLR
  H1 Value                          -11322.435  → LL for a saturated (perfect) model
  H1 Scaling Correction Factor          1.4073  → indicates how far off from 1=normal
      for MLR

Information Criteria → "smaller is better" – use for nested or non-nested model comparisons

   Akaike (AIC)                      23108.808  → AIC = (-2*LL_H0) + (2*estimated parameters)
   Bayesian (BIC)                    23198.912  → BIC = (-2*LL_H0) + (LN N*estimated parameters)
   Sample-Size Adjusted BIC          23141.739  → BIC replacing N with (N + 2) / 24
     (n* = (n + 2) / 24)

Chi-Square Test of Model Fit (Significance is bad) → rescaled LRT for your model vs saturated

    Value                                307.799
    Degrees of Freedom                         9 → # parameters leftover = Model DF
    P-Value                               0.0000
    Scaling Correction Factor             1.3903 → indicates how far off from normal=1
        for MLR                                    > 1 = leptokurtic distribution (too-fat tails)
                                                   < 1 = platykurtotic distribution (too-thin tails)

*   The chi-square value for MLM, MLMV, MLR, ULSMV, WLSM and WLSMV cannot be used
    for chi-square difference testing in the regular way.  MLM, MLR and WLSM
    chi-square difference testing is described on the Mplus website.  MLMV, WLSMV,
    and ULSMV difference testing is done using the DIFFTEST option.
```

Where does this χ^2 value for "model fit" come from? A rescaled −2LL model comparison of this one-factor model (H0) against the saturated model (H1) that perfectly reproduces the data covariances:

Step 1: Original −2ΔLL = −2*(LL_{fewer} – LL_{more}) = −2(−11,536.404 + 11,322.435) = 427.938

Step 2: Scaling correction = [(#$parms_{fewer}$*$scale_{fewer}$) – (#$parms_{more}$*$scale_{more}$)] / (#$parms_{fewer}$ – #$parms_{more}$)
= [(18 * 1.4159) – (27 * 1.4073)] / (18 – 27) = −12.501 / −9 = 1.3903

Step 3: Rescaled −2ΔLL = −2ΔLL / scaling correction = 427.938 / 1.903 = **307.847** → ~matches model χ^2

Step 4: Difference in df = #$parms_{more}$ – #$parms_{fewer}$ = 27 – 18 = **9**

```
RMSEA (Root Mean Square Error Of Approximation)→ how much worse than 0 = saturated model

    Estimate                              0.173 → smaller is better
    90 Percent C.I.              0.157    0.190 → CI will become smaller as sample size increases
    Probability RMSEA <= .05              0.000 → so RMSEA does NOT overlap .05 (is signif > .05)


CFI/TLI → how much better than 0 = null model (bigger is better)

    CFI                                   0.732
    TLI                                   0.553
```

```
SRMR (Standardized Root Mean Square Residual)→ how much worse than 0 = saturated model

   Value                             0.086

Chi-Square Test of Model Fit for the Baseline Model → LRT of null vs saturated (who cares)

     Value                             1128.693
     Degrees of Freedom                      15
     P-Value                             0.0000
```

Where does this χ^2 value for "fit of the baseline model" come from? A rescaled −2LL model comparison of the independence "null" model with NO covariances to the saturated model:

Step 1: Original $-2\Delta LL = -2*(LL_{fewer} - LL_{more}) = -2(-12{,}312.952 + 11{,}322.435) = 1{,}981.034$

Step 2: Scaling correction $= [\ (\#parms_{fewer}*scale_{fewer}) - (\#parms_{more}*scale_{more})\ ] / (\#parms_{fewer} - \#parms_{more})$
$= [\ (12 * 0.9725) - (27 * 1.4073)\ ] / (12 - 27) = -26.372 / -15 = 1.7551$

Step 3: Rescaled $-2\Delta LL = -2\Delta LL$ / scaling correction = 1,981.034 / 1.7551 = **1,128.704** → ~matches baseline χ^2
Step 4: Difference in df $= \#parms_{more} - \#parms_{fewer} = 27 - 12 =$ **15**

What's the point? This baseline model fit test tells us whether there are any covariances at all (i.e., whether it even makes sense to try to fit latent factors to predict the item covariances).

R model fit information for a single-factor model (same regardless of factor scaling method):

```
Model Test User Model:                        For ML:     For MLR:
                                             Standard      Robust
  Test Statistic                              427.937     307.803 → LRT for your model vs H1
  Degrees of freedom                                9           9 → # parms leftover = Model DF
  P-value (Chi-square)                          0.000       0.000
  Scaling correction factor                                 1.390 → how far off from normal=1

    Yuan-Bentler correction (Mplus variant)

Model Test Baseline Model:
  Test statistic                             1981.034    1128.693 → LRT of null vs saturated
  Degrees of freedom                               15          15
  P-value                                       0.000       0.000
  Scaling correction factor                                 1.755

User Model versus Baseline Model: → how much better than 0 = null model (bigger is better)

  Comparative Fit Index (CFI)                   0.787       0.732
  Tucker-Lewis Index (TLI)                      0.645       0.553

  Robust Comparative Fit Index (CFI)                        0.787 → not in Mplus
  Robust Tucker-Lewis Index (TLI)                           0.646 → not in Mplus

Loglikelihood and Information Criteria: → For LL, bigger is better; for IC, smaller is better

  Loglikelihood user model (H0)            -11536.404  -11536.404 → LL for your model
  Scaling correction factor                                 1.416 → how far off from 1=normal

      for the MLR correction
  Loglikelihood unrestricted model (H1)    -11322.435  -11322.435 → LL for saturated model
  Scaling correction factor                                 1.407 → how far off from 1=normal
      for the MLR correction

  Akaike (AIC)                              23108.808   23108.808 → smaller is better
  Bayesian (BIC)                            23198.912   23198.912
  Sample-size adjusted Bayesian (BIC)       23141.739   23141.739
```

```
Root Mean Square Error of Approximation: → how much worse than 0 = saturated model

  RMSEA                                         0.205       0.173 → smaller is better
  90 Percent confidence interval - lower        0.189       0.160
  90 Percent confidence interval - upper        0.222       0.188
  P-value RMSEA <= 0.05                         0.000       0.000

  Robust RMSEA                                              0.205 → not in Mplus
  90 Percent confidence interval - lower                    0.185
  90 Percent confidence interval - upper                    0.224

Standardized Root Mean Square Residual: → how much worse than 0 = saturated model

  SRMR                                          0.086       0.086
```

FYI for demonstration purposes, here is how to fit the saturated (Unstructured) Baseline Model: Item means, variances, and covariances in original data

```
! Saturated Model for Demonstration Purposes
! All item means, variances, and covariances estimated
    ! Item means --> [ ] indicates means or intercepts, @=fixed, *=free
      [Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*];
    ! Item variances --> just list item by itself, @=fixed, *=free
      Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;
    ! Item covariances --> just list all by all, @=fixed, *=free
      Sit1r  Sit2  Sit3r  Sit4  Sit5r  Sit6  WITH
      Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;
```

```
# R Syntax for lavaan function: longer but more transparent version of model
  SyntaxSat = "
# Item means all estimated
  Sit1r ~ 1; Sit2 ~ 1; Sit3r  ~ 1; Sit4 ~ 1; Sit5r ~ 1; Sit6 ~ 1
# Item variances all estimated
  Sit1r ~~ Sit1r; Sit2 ~~ Sit2; Sit3r ~~ Sit3r; Sit4 ~~ Sit4; Sit5r ~~ Sit5r; Sit6 ~~ Sit6
# Covariances all estimated
  Sit1r ~~ Sit2; Sit1r ~~ Sit3r; Sit1r ~~ Sit4; Sit1r ~~ Sit5r; Sit1r ~~ Sit6
  Sit2 ~~ Sit3r; Sit2 ~~ Sit4; Sit2 ~~ Sit5r; Sit2 ~~ Sit6; Sit3r ~~ Sit4; Sit3r ~~ Sit5r
  Sit3r ~~ Sit6; Sit4 ~~ Sit5r; Sit4 ~~ Sit6; Sit5r ~~ Sit6
"
ModelSat = lavaan(model=SyntaxSat, data=Example4, estimator="MLR", mimic="mplus")
summary(ModelSat, fit.measures=TRUE, standardized=TRUE)
# Get saturated model-implied means, variances, and covariances
  fitted(object=ModelSat)
```

Model fit information for the <u>saturated model</u>: illustrating what the χ^2 test of global model fit means

```
Number of Free Parameters                      27 → all possible means, variances, covariances

Loglikelihood
          H0 Value                       -11322.435
          H0 Scaling Correction Factor       1.4073
            for MLR
          H1 Value                       -11322.435
          H1 Scaling Correction Factor       1.4073
            for MLR

Information Criteria
          Akaike (AIC)                    22698.870
          Bayesian (BIC)                  22834.027
          Sample-Size Adjusted BIC        22748.268
            (n* = (n + 2) / 24)

Chi-Square Test of Model Fit
          Value                              0.000*
          Degrees of Freedom                     0
          P-Value                           0.0000
          Scaling Correction Factor         1.0000
            for MLR
```

Note that H0 and H1 are now the same!
Our H0 model = the H1 saturated model.

```
From R: > fitted(object = ModelSat)
$cov
      Sit1r Sit2  Sit3r Sit4  Sit5r Sit6
Sit1r 3.049
Sit2  0.577 1.903
Sit3r 1.802 0.697 2.543
Sit4  0.734 1.103 0.824 1.967
Sit5r 1.358 0.604 1.319 0.695 2.945
Sit6  0.795 0.965 0.868 0.962 0.798 2.341

$mean
Sit1r  Sit2 Sit3r  Sit4 Sit5r  Sit6
4.547 5.289 4.896 5.359 4.860 5.321
```

As another FYI for demonstration purposes, here is how to fit the Independence (Null) Baseline Model: Item means and variances, but NO covariances

```
! Null Independence Model for Demonstration Purposes
! All means and variances estimated, all covariances fixed=0
    ! Item means --> [ ] indicates means or intercepts, @=fixed, *=free
      [Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*];
    ! Item variances --> just list item by itself, @=fixed, *=free
      Sit1r* Sit2* Sit3r* Sit4* Sit5r* Sit6*;
    ! NO Item covariances --> just list all by all, @=fixed to 0
      Sit1r  Sit2  Sit3r  Sit4  Sit5r  Sit6  WITH
      Sit1r@0 Sit2@0 Sit3r@0 Sit4@0 Sit5r@0 Sit6@0;
```

```
# R Syntax for lavaan function: longer but more transparent version of model
  SyntaxNull = "
# Item means all estimated
  Sit1r ~ 1; Sit2 ~ 1; Sit3r  ~ 1; Sit4 ~ 1; Sit5r ~ 1; Sit6 ~ 1
# Item variances all estimated
  Sit1r ~~ Sit1r; Sit2 ~~ Sit2; Sit3r ~~ Sit3r; Sit4 ~~ Sit4; Sit5r ~~ Sit5r; Sit6 ~~ Sit6
"
ModelNull = lavaan(model=SyntaxNull, data=Example4, estimator="MLR", mimic="mplus")
summary(ModelNull, fit.measures=TRUE)
# Get null model-implied means, variances, and covariances
  fitted(object=ModelNull)
```

Model fit for the <u>independence "null" model:</u> illustrating what RMSEA, CFI, and TLI mean

```
Number of Free Parameters                        12

Loglikelihood
          H0 Value                       -12312.952
          H0 Scaling Correction Factor       0.9725
            for MLR
          H1 Value                       -11322.435
          H1 Scaling Correction Factor       1.4073
            for MLR

Information Criteria
          Akaike (AIC)                    24649.904
          Bayesian (BIC)                  24709.974
          Sample-Size Adjusted BIC        24671.859
            (n* = (n + 2) / 24)

Chi-Square Test of Model Fit
          Value                            1128.692*
          Degrees of Freedom                     15
          P-Value                            0.0000
          Scaling Correction Factor          1.7552
            for MLR

RMSEA (Root Mean Square Error Of Approximation)
          Estimate                            0.259
          90 Percent C.I.              0.247  0.272
          Probability RMSEA <= .05            0.000

CFI/TLI
          CFI                                 0.000
          TLI                                 0.000

Chi-Square Test of Model Fit for the Baseline Model
          Value                            1128.693
          Degrees of Freedom                     15
          P-Value                            0.0000

SRMR (Standardized Root Mean Square Residual)
          Value                               0.300
```

```
From R: > fitted(object = ModelNull)
$cov
      Sit1r Sit2  Sit3r Sit4  Sit5r Sit6
Sit1r 3.049
Sit2  0.000 1.903
Sit3r 0.000 0.000 2.543
Sit4  0.000 0.000 0.000 1.967
Sit5r 0.000 0.000 0.000 0.000 2.945
Sit6  0.000 0.000 0.000 0.000 0.000 2.341

$mean
Sit1r  Sit2 Sit3r  Sit4 Sit5r  Sit6
4.547 5.289 4.896 5.359 4.860 5.321
```

Note that the model fit is the same as the "baseline" model fit given before.

Although not 0, this is the worst possible RMSEA while still allowing separate means and variances per item in these data. RMSEA is a parsimony-corrected absolute fit index (so, its fit is relative to the saturated model).

CFI and TLI are 0 because they are "incremental fit" indices relative to the independence model (which this is).

SRMR is also an absolute fit index (relative to saturated model), so this is the worst it gets for these data, too.

Here is <u>what the single-factor model implied</u> for our item means, variances, and covariances:

Mplus: scroll to RESIDUAL OUTPUT → info in “Model Estimated” sections

R:
```
# R Get model-implied means, variances, and covariances
  fitted(object=Model1)

$cov → Model-implied matrix
      Sit1r Sit2  Sit3r Sit4  Sit5r Sit6
Sit1r 3.049
Sit2  0.867 1.903
Sit3r 1.531 0.872 2.543
Sit4  0.968 0.551 0.973 1.967
Sit5r 1.263 0.719 1.269 0.803 2.945
Sit6  1.011 0.575 1.016 0.642 0.838 2.341

$mean → Model-implied means
Sit1r  Sit2 Sit3r  Sit4 Sit5r  Sit6
4.547 5.289 4.896 5.359 4.860 5.321
```

VS Saturated Model `fitted(object=ModelSat)`
```
$cov → From original data
      Sit1r Sit2  Sit3r Sit4  Sit5r Sit6
Sit1r 3.049
Sit2  0.577 1.903
Sit3r 1.802 0.697 2.543
Sit4  0.734 1.103 0.824 1.967
Sit5r 1.358 0.604 1.319 0.695 2.945
Sit6  0.795 0.965 0.868 0.962 0.798 2.341

$mean → From original data
Sit1r  Sit2 Sit3r  Sit4 Sit5r  Sit6
4.547 5.289 4.896 5.359 4.860 5.321
```

But global fit for the one-factor model is not good enough—let’s examine the residual info, which provides the <u>discrepancy</u> between what the model implies and what the H1 data model says:

```
# R Get residual (discrepancy) info for local misfit in covariance and correlation form
  resid(object=Model1, type="raw"); resid(object=Model1, type="cor")

R $cov → Discrepancies in means, variances, and covariances
      Sit1r  Sit2   Sit3r  Sit4   Sit5r  Sit6
Sit1r  0.000
Sit2  -0.290  0.000
Sit3r  0.271 -0.174  0.000
Sit4  -0.234  0.552 -0.149  0.000
Sit5r  0.096 -0.114  0.050 -0.108  0.000
Sit6  -0.216  0.390 -0.149  0.319 -0.040  0.000

$mean
Sit1r  Sit2 Sit3r  Sit4 Sit5r  Sit6
    0     0     0     0     0     0

R $cov → Discrepancies in correlations (STDYX)
      Sit1r  Sit2   Sit3r  Sit4   Sit5r  Sit6
Sit1r  0.000
Sit2  -0.120  0.000
Sit3r  0.097 -0.079  0.000
Sit4  -0.096  0.285 -0.067  0.000
Sit5r  0.032 -0.048  0.018 -0.045  0.000
Sit6  -0.081  0.185 -0.061  0.149 -0.015  0.000
```

All values are from: observed – predicted

Top: The variances and means have discrepancies=0 → perfect recovery, which means they are not the cause of our bad fit. **Thus, misfit results from the difference between the observed and model-implied (model-recreated) covariances.**

Bottom: Although this is called the “residuals for correlations” matrix in Mplus, it is NOT the same as “residual correlations”, which are error correlations as model parameters. These are **discrepancies** in correlation!

From Mplus: Residuals for Correlations (Observed – Predicted Correlations)

	SIT1R	SIT2	SIT3R	SIT4	SIT5R	SIT6
SIT1R	0.000					
SIT2	**-0.120**	0.000				
SIT3R	0.097	-0.079	0.000			
SIT4	-0.096	**0.285**	-0.067	0.000		
SIT5R	0.032	-0.048	0.018	-0.045	0.000	
SIT6	-0.081	**0.185**	-0.061	**0.149**	-0.015	0.000

We can also get “normalized” residuals, which can be thought of as z-scores for how large the residual leftover covariance is in absolute terms. Because the denominator decreases with sample size, however, these values may be inflated in large samples, so watch for *relatively* large values.

```
# R: Get normalized residuals (like z-scores for discrepancies)
  resid(object=Model1, type="normalized")
```

"Normalized" Residuals for Inter-Item Covariances: z = (observed – predicted) / SE(observed)

Mplus Normalized Residuals for Covariances (same results in R)

	SIT1R	SIT2	SIT3R	SIT4	SIT5R	SIT6
SIT1R	0.000					
SIT2	**-3.503**	0.000				
SIT3R	2.977	-2.253	0.000			
SIT4	-2.928	**6.560**	-1.959	0.000		
SIT5R	0.960	-1.434	0.548	-1.372	0.000	
SIT6	-2.345	**4.721**	-1.756	**3.925**	-0.444	0.000

NEGATIVE NORMALIZED RESIDUAL → **Less related** than you predicted (want to be **less** together)
POSITIVE NORMALIZED RESIDUAL → **More related** than you predicted (want to be **more** together)

Why might the normalized residuals (leftover correlations) for the positive-worded items be larger than for the negatively-worded items?

What do the "cheat code" modification indices suggest we do to improve model fit?

MPLUS: MODEL MODIFICATION INDICES (MI = change in model test chi-square)
Minimum M.I. value for printing the modification index 6.635
EPC = EXPECTED PARAMETER CHANGE

WITH Statements (SUGGESTED ERROR COVARIANCES for unaccounted for multidimensionality)

			M.I.	E.P.C.	Std E.P.C.	StdYX E.P.C.
SIT2	WITH	SIT1R	49.618	-0.464	-0.464	-0.316
SIT3R	**WITH**	**SIT1R**	**143.624**	**1.023**	**1.023**	**0.827**
SIT3R	WITH	SIT2	34.877	-0.357	-0.357	-0.300
SIT4	WITH	SIT1R	36.280	-0.403	-0.403	-0.280
SIT4	**WITH**	**SIT2**	**161.318**	**0.702**	**0.702**	**0.509**
SIT4	WITH	SIT3R	29.202	-0.336	-0.336	-0.288
SIT6	WITH	SIT1R	24.079	-0.358	-0.358	-0.224
SIT6	**WITH**	**SIT2**	**63.893**	**0.486**	**0.486**	**0.317**
SIT6	WITH	SIT3R	22.386	-0.319	-0.319	-0.246
SIT6	WITH	SIT4	46.541	0.415	0.415	0.276

```
# Get "cheat code" modification indices
  modificationindices(object=Model4, sort.=TRUE)
```

```
> modificationindices(object = Model1, sort. = TRUE)
     lhs op   rhs      mi    epc sepc.lv sepc.all sepc.nox
27  Sit2 ~~  Sit4 224.295  0.702   0.702    0.509    0.509
22 Sit1r ~~ Sit3r 199.679  1.023   1.023    0.827    0.827
29  Sit2 ~~  Sit6  88.835  0.486   0.486    0.317    0.317
21 Sit1r ~~  Sit2  68.985 -0.464  -0.464   -0.316   -0.316
34  Sit4 ~~  Sit6  64.710  0.415   0.415    0.276    0.276
23 Sit1r ~~  Sit4  50.442 -0.403  -0.403   -0.280   -0.280
26  Sit2 ~~ Sit3r  48.505 -0.357  -0.357   -0.300   -0.300
30 Sit3r ~~  Sit4  40.616 -0.336  -0.336   -0.288   -0.288
25 Sit1r ~~  Sit6  33.475 -0.358  -0.358   -0.224   -0.224
32 Sit3r ~~  Sit6  31.132 -0.319  -0.319   -0.246   -0.246
28  Sit2 ~~ Sit5r   7.117 -0.151  -0.151   -0.092   -0.092
33  Sit4 ~~ Sit5r   6.906 -0.149  -0.149   -0.093   -0.093
24 Sit1r ~~ Sit5r   6.417  0.176   0.176    0.104    0.104
31 Sit3r ~~ Sit5r   3.548  0.123   0.123    0.089    0.089
35 Sit5r ~~  Sit6   0.736 -0.053  -0.053   -0.030   -0.030
```

Here the MI column provides the **non-robust (regular ML) version of the change in model test chi-square** from adding that parameter. Lavaan used to provide the robust version as mi.scaled, but not anymore…?

But we come to the same conclusion either way about which new parameters would result in the biggest change in fit… note that all suggestions are error covariances! This is not a coincidence, because **these cheat codes will never suggest a new model**!

These results suggest that wording valence is playing a larger role in the pattern of covariance across items than what the one-factor model predicts. Rather than adding many covariances among the residuals for specific items, how about <u>a two-factor model based on wording instead</u>?

Model 4. Model with Two Fully Z-Scored Factors ***(also adding labels in parentheses to the end of each parameter to use in computing omega reliability per factor)***

```
! Mplus Model 4: Two Factors Using Fully Z-Scored Factor Scaling with all parameters labeled
    SitP BY Sit2*  Sit4*  Sit6*  (L1-L3);       ! SitP loadings (all free)
    SitN BY Sit1r* Sit3r* Sit5r* (L4-L6);       ! SitN loadings (all free)
    [Sit2*  Sit4*  Sit6*]    (I1-I3);           ! SitP intercepts (all free)
    [Sit1r* Sit3r* Sit5r*]   (I4-I6);           ! SitN intercepts (all free)
    Sit2*   Sit4*  Sit6*     (E1-E3);           ! SitP residual variances (all free)
    Sit1r*  Sit3r* Sit5r*    (E4-E6);           ! SitN residual variances (all free)
    SitP@1 (VarP); SitN@1    (VarN);            ! Factor variances (fixed=1)
    SitP WITH SitN*  (FactCov);                 ! Factor covariance (free)
    [SitP@0 SitN@0]  (MeanP MeanN);             ! Factor means (fixed=0)

MODEL CONSTRAINT:   ! Mplus syntax to compute omega sum score reliability per factor
NEW(OmegaP OmegaN); ! Using 1 as placeholder for factor variances
    OmegaP = (1*(L1+L2+L3)**2) / ((1*(L1+L2+L3)**2) + (E1+E2+E3));
    OmegaN = (1*(L4+L5+L6)**2) / ((1*(L4+L5+L6)**2) + (E4+E5+E6));
! Don't forget to include any error covariances, too! (see Lecture 4b slide 47)
```

```
# R Syntax for lavaan function: longer but more transparent version of model
Syntax4 = "  ! Note labels are in order of inclusion, not according to item name
# Define factor and request item factor loadings all estimated
  SitP =~ L1*Sit2  + L2*Sit4  + L3*Sit6     # Pos items
  SitN =~ L4*Sit1r + L5*Sit3r + L6*Sit5r    # Neg items
# Item intercepts all estimated
  Sit2  ~ I1*1; Sit4  ~ I2*1; Sit6  ~ I3*1  # Pos items
  Sit1r ~ I4*1; Sit3r ~ I5*1; Sit5r ~ I6*1  # Neg items
# Item error variances all estimated
  Sit2  ~~ E1*Sit2;  Sit4  ~~ E2*Sit4;  Sit6  ~~ E3*Sit6   # Pos items
  Sit1r ~~ E4*Sit1r; Sit3r ~~ E5*Sit3r; Sit5r ~~ E6*Sit5r  # Neg items
# Factor variances fixed=1 and factor means fixed=0 (won't allow labels)
  SitP ~~ 1*SitP; SitN ~~ 1*SitN; SitP ~ 0; SitN ~ 0
# Factor covariance estimated (and labeled)
  SitP ~~ FactCov*SitN
# Calculate Omega Reliability for Sum Score Per Factor (1=factor variance):
  OmegaP := (1*(L1 + L2 + L3)^2) / ( (1*(L1 + L2 + L3)^2) + (E1 + E2 + E3))
  OmegaN := (1*(L4 + L5 + L6)^2) / ( (1*(L4 + L5 + L6)^2) + (E4 + E5 + E6))
# Don't forget to include any error covariances, too! (see Lecture 4b slide 47)
"
# Use MLR estimation like in Mplus, z-score latent factors (mean=0, SD=1)
Model4 = lavaan(model=Syntax4, data=Example4, estimator="MLR", mimic="mplus", std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution
summary(object=Model4, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

print("LRT for one factor vs two factors: is correlation < 1?")
anova(Model1, Model4)

# Get model-implied means, variances, and covariances
fitted(object=Model4)
# Get residual (discrepancy) info for local misfit in covariance and correlation form
resid(object=Model4, type="raw"); resid(object=Model4, type="cor")
# Get normalized residuals (like z-scores for discrepancies)
resid(object=Model4, type="normalized")
# Get "cheat code" modification indices
modificationindices(object = Model4, sort.=TRUE)
```

```
# R Syntax for sem or cfa functions: shorter but less transparent version of model
Syntax4short = "
  SitP =~ L1*Sit2  + L2*Sit4  + L3*Sit6     # Pos items
  SitN =~ L4*Sit1r + L5*Sit3r + L6*Sit5r    # Neg items
"
# Use MLR estimation like in Mplus, z-score latent factor (mean=0, SD=1)
Model4short = sem(model=Syntax4short, data=Example4, estimator="MLR", mimic="mplus",
                  std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution (will be same as from lavaan)
summary(object=Model4short, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

```
MPLUS MODEL FIT INFORMATION

Number of Free Parameters                       19

Loglikelihood
    H0 Value                            -11340.140
    H0 Scaling Correction Factor            1.4017
         for MLR
    H1 Value                            -11322.435
    H1 Scaling Correction Factor            1.4073
         for MLR

Information Criteria
    Akaike (AIC)                         22718.281
    Bayesian (BIC)                       22813.391
    Sample-Size Adjusted BIC             22753.042
       (n* = (n + 2) / 24)

Chi-Square Test of Model Fit
    Value                                   24.924*
    Degrees of Freedom                           8
    P-Value                                 0.0016
    Scaling Correction Factor               1.4207
         for MLR

RMSEA (Root Mean Square Error Of Approximation)
    Estimate                                 0.044
    90 Percent C.I.                  0.025   0.064
    Probability RMSEA <= .05                 0.667

CFI/TLI
    CFI                                      0.985
    TLI                                      0.972

Chi-Square Test of Model Fit for the Baseline Model
    Value                                 1128.693
    Degrees of Freedom                          15
    P-Value                                 0.0000

SRMR (Standardized Root Mean Square Residual)
    Value                                    0.029
```

Is the 2-factor model better than the 1-factor model? How do we know?

Rescaled likelihood ratio test between alternative H0 models (−2LL rescaled difference test):

1. **−2ΔLL = −2* difference in LL:**
   −2*(−11,536.404 − −11,340.140) = 392.528

2. **difference scaling correction:**
   $(\text{parms}_1\text{*scale}_1) - (\text{parms}_2\text{*scale}_2) / (\text{parms}_1 - \text{parms}_2)$
   (18*1.4158) – (19*1.4017) / (18 – 19) = 1.1479

3. **rescaled difference = −2ΔLL / scaling correction:**
   392.528 / 1.1479 = 341.953

4. **compare rescaled difference to χ^2 with DF = ΔDF:**
   critical χ^2 for DF =1 is 3.84, so because 341.953 is > 3.84, the model fit significantly improved

Report LRT as: −2ΔLL(1) = 342, $p < .001$
Or as: $\chi^2(1) = 342$, $p < .001$

FILL IN				CALCULATED				
	Test of -2ΔLL Difference							
Models: Fewer Parms in Row 1 More Parms in Row 2	Model H0 LL	H0 LL Scale Factor	# Free Parms	Diff in LL * -2	Diff Scaling Correction	Scaled Diff in -2LL	DF Diff	Exact P-Value
Demonstration of LRTs you have to do yourself (alternative H0 models)								
One-Factor	-11,536.404	1.4158	18					
Two-Factor	-11,340.140	1.4017	19					
Test of Difference				392.528	1.1479	341.953	1	0.0000

Model comparison from lavaan using anova:

```
[1] "LRT for one factor vs two factors: is correlation < 1?"
> anova(Model1, Model4)
Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
lavaan NOTE:
    The "Chisq" column contains standard test statistics, not the
    robust test that should be reported per model. A robust difference
    test is a function of two standard (not robust) statistics.

       Df     AIC     BIC    Chisq Chisq diff Df diff Pr(>Chisq)
Model4  8 22718.3 22813.4  35.4104
Model1  9 23108.8 23198.9 427.9371    342.289       1 < 2.22e-16
```

MPLUS UNSTANDARDIZED RESULTS

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value
SITP BY				
SIT2	1.007	0.052	19.487	0.000
SIT4	1.064	0.050	21.195	0.000
SIT6	0.956	0.053	18.203	0.000
SITN BY				
SIT1R	1.325	0.048	27.698	0.000
SIT3R	1.349	0.044	30.514	0.000
SIT5R	1.009	0.055	18.358	0.000
SITP WITH SITN = factor covariance (= correlation if variances=1)				
	0.564	**0.041**	**13.776**	**0.000**
Means				
SITP	0.000	0.000	999.000	999.000
SITN	0.000	0.000	999.000	999.000
Intercepts				
SIT1R	4.547	0.053	86.474	0.000
SIT2	5.289	0.042	127.347	0.000
SIT3R	4.896	0.048	101.959	0.000
SIT4	5.359	0.042	126.896	0.000
SIT5R	4.860	0.052	94.060	0.000
SIT6	5.321	0.046	115.492	0.000
Variances				
SITP	1.000	0.000	999.000	999.000
SITN	1.000	0.000	999.000	999.000
Residual Variances				
SIT1R	1.294	0.103	12.547	0.000
SIT2	0.888	0.097	9.173	0.000
SIT3R	0.724	0.092	7.857	0.000
SIT4	0.835	0.093	9.003	0.000
SIT5R	1.926	0.119	16.128	0.000
SIT6	1.428	0.134	10.684	0.000
New/Additional Parameters				
OMEGAP	0.744	0.020	37.956	0.000
OMEGAN	0.775	0.014	56.803	0.000

<u>Omega</u> =

Var(Factor) * (Sum of loadings)^2 /
Var(Factor)* (Sum of loadings)^2 +
Sum of error variances +
2* Sum of error covariances

<u>Omega for Positive Factor = .744</u>
1.0*(1.007+1.064+0.956)^2 /
1.0*(1.007+1.064+0.956)^2 +
(0.888+0.835+1.428) + 2*0

(alpha was .746, btw)

<u>Omega for Negative Factor = .775</u>
1.0*(1.325+1.349+1.009)^2 /
1.0*(1.325+1.349+1.009)^2 +
(1.294+0.724+1.926) + 2*0

(alpha was .780, btw)

STDYX STANDARDIZED RESULTS

	Estimate	S.E.	Est./S.E.	Two-Tailed P-Value
SITP BY				
SIT2	0.730	0.032	22.794	0.000
SIT4	0.759	0.029	25.995	0.000
SIT6	0.625	0.035	17.949	0.000
SITN BY				
SIT1R	0.759	0.022	34.072	0.000
SIT3R	0.846	0.021	39.657	0.000
SIT5R	0.588	0.030	19.651	0.000
SITP WITH				
SITN	0.564	0.041	13.776	0.000
Residual Variances				
SIT1R	0.425	0.034	12.567	0.000
SIT2	0.467	0.047	9.976	0.000
SIT3R	0.285	0.036	7.895	0.000
SIT4	0.425	0.044	9.589	0.000
SIT5R	0.654	0.035	18.576	0.000
SIT6	0.610	0.043	14.029	0.000
R-SQUARE				
SIT1R	0.575	0.034	17.036	0.000
SIT2	0.533	0.047	11.397	0.000
SIT3R	0.715	0.036	19.829	0.000
SIT4	0.575	0.044	12.998	0.000
SIT5R	0.346	0.035	9.826	0.000
SIT6	0.390	0.043	8.974	0.000

R Output, which adds the labels you gave to the parameters (a nice feature):

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP =~
    Sit2      (L1)    1.007    0.052   19.487    0.000    1.007    0.730
    Sit4      (L2)    1.064    0.050   21.195    0.000    1.064    0.759
    Sit6      (L3)    0.956    0.053   18.203    0.000    0.956    0.625
  SitN =~
    Sit1r     (L4)    1.325    0.048   27.698    0.000    1.325    0.759
    Sit3r     (L5)    1.349    0.044   30.514    0.000    1.349    0.846
    Sit5r     (L6)    1.009    0.055   18.358    0.000    1.009    0.588

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP ~~
    SitN    (FctC)    0.564    0.041   13.775    0.000    0.564    0.564

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2      (I1)    5.289    0.042  127.346    0.000    5.289    3.834
   .Sit4      (I2)    5.359    0.042  126.896    0.000    5.359    3.821
   .Sit6      (I3)    5.321    0.046  115.492    0.000    5.321    3.477
   .Sit1r     (I4)    4.547    0.053   86.474    0.000    4.547    2.604
   .Sit3r     (I5)    4.896    0.048  101.959    0.000    4.896    3.070
   .Sit5r     (I6)    4.860    0.052   94.060    0.000    4.860    2.832
    SitP              0.000                               0.000    0.000
    SitN              0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2      (E1)    0.888    0.097    9.173    0.000    0.888    0.467
   .Sit4      (E2)    0.835    0.093    9.003    0.000    0.835    0.425
   .Sit6      (E3)    1.428    0.134   10.684    0.000    1.428    0.610
   .Sit1r     (E4)    1.294    0.103   12.547    0.000    1.294    0.425
   .Sit3r     (E5)    0.724    0.092    7.857    0.000    0.724    0.285
   .Sit5r     (E6)    1.926    0.119   16.128    0.000    1.926    0.654
    SitP              1.000                               1.000    1.000
    SitN              1.000                               1.000    1.000

R-Square:
                   Estimate
    Sit2              0.533
    Sit4              0.575
    Sit6              0.390
    Sit1r             0.575
    Sit3r             0.715
    Sit5r             0.346

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    OmegaP            0.744    0.020   37.956    0.000    0.744    0.748
    OmegaN            0.775    0.014   56.802    0.000    0.775    0.779
```

The two-factor model appears to fit well overall, but is there any remaining sizeable local misfit?

```
# R Get model-implied means, variances, and covariances
fitted(object=Model4)
# R Get residual (discrepancy) info for local misfit in covariance and correlation form
resid(object=Model4, type="raw"); resid(object=Model4, type="cor")
# R Get normalized residuals (like z-scores for discrepancies)
resid(object=Model4, type="normalized")
# R Get "cheat code" modification indices
modificationindices(object = Model4, sort.=TRUE)
```

Mplus residuals of correlation matrix (observed – model-implied correlation):

	SIT1R	SIT2	SIT3R	SIT4	SIT5R	SIT6
SIT1R	0.000					
SIT2	**-0.073**	0.000				
SIT3R	0.006	-0.031	0.000			
SIT4	-0.025	0.016	0.007	0.000		
SIT5R	0.007	0.013	-0.015	0.037	0.000	
SIT6	0.030	0.001	0.057	-0.026	**0.097**	0.000

Mplus "Normalized" residuals (z-like statistic for how far off each covariance is):

	SIT1R	SIT2	SIT3R	SIT4	SIT5R	SIT6
SIT1R	0.000					
SIT2	**-2.125**	0.000				
SIT3R	0.172	-0.896	0.000			
SIT4	-0.768	0.370	0.192	0.000		
SIT5R	0.212	0.382	-0.464	1.128	0.000	
SIT6	0.869	0.031	1.658	-0.676	**2.847**	0.000

Any suggested cheat codes? (only available in Mplus when <u>not</u> using MODEL CONSTRAINT)

```
MPLUS MODEL MODIFICATION INDICES
Minimum M.I. value for printing the modification index    6.635

                            M.I.     E.P.C.  Std E.P.C.  StdYX E.P.C.

BY Statements - these are cross-loadings

SITN      BY SIT2            9.775    -0.224      -0.224       -0.162
SITN      BY SIT6           10.828     0.245       0.245        0.160

WITH Statements - these are error covariances

SIT4      WITH SIT2         10.830     0.332       0.332        0.386
SIT6      WITH SIT4          9.773    -0.273      -0.273       -0.250

> modificationindices(object = Model4, sort. = TRUE)
     lhs op   rhs     mi    epc sepc.lv sepc.all sepc.nox
31  SitN =~  Sit6 15.383  0.245   0.245    0.160    0.160
32  Sit2 ~~  Sit4 15.383  0.332   0.332    0.386    0.386
37  Sit4 ~~  Sit6 13.886 -0.273  -0.273   -0.250   -0.250
29  SitN =~  Sit2 13.885 -0.224  -0.224   -0.162   -0.162
26  SitP =~  Sit1r  9.244 -0.223  -0.223   -0.128   -0.128
46 Sit3r ~~  Sit5r  9.244 -0.277  -0.277   -0.234   -0.234
```

=~ are cross-loadings
~~ are error covariances

Because we have no real theoretical or defendable reason to fit any of these suggested parameters, we will not add any new parameters. This will be about as good as it gets.

Let's examine the estimated distribution of the factor scores for each factor… these are EAP (expected a posteriori) estimates, which are the <u>mean</u> of each person's random factor distribution

To do so in Mplus: turn on SAVE and PLOT options in syntax; open *Plot* menu in output, select *View plots*, select *histograms*, select *view*, in *plot properties* tab use the drop-down menu to select your factor name, then go to *display properties* tab and select *histogram/density plot*. When the plot appears, you can customize the axes using the menus visible by right-clicking on the plot.

To do so in R: see example code online, with more details in this handout:
https://jonathantemplin.com/wp-content/uploads/2017/09/EPSY906_Example04.nb_.html#

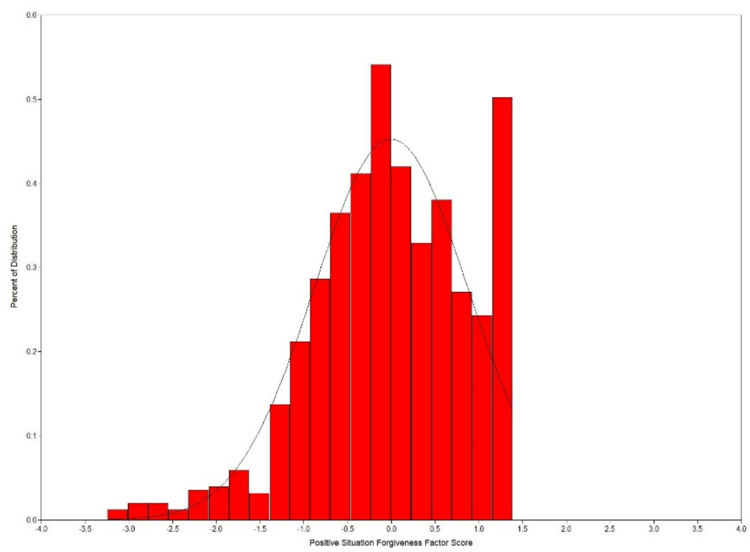
```
# Get factor scores
Model4Scores = lavPredict(object=Model4, newdata=NULL, type="lv", method="EBM",
                          se="standard", acov="standard", label=TRUE, fsm=FALSE, append.data=TRUE)
```

```
MPLUS SAMPLE STATISTICS FOR ESTIMATED FACTOR SCORES
     SAMPLE STATISTICS
           Means
              SITP          SITP_SE       SITN          SITN_SE
              ________      ________      ________      ________
 1              0.000         0.472         0.000         0.418

           Covariances
              SITP          SITP_SE       SITN          SITN_SE
              ________      ________      ________      ________
 SITP           0.777
 SITP_SE        0.000         0.000
 SITN           0.533         0.000         0.825
 SITN_SE        0.000         0.000         0.000         0.000

           Correlations
              SITP          SITP_SE       SITN          SITN_SE
              ________      ________      ________      ________
 SITP           1.000
 SITP_SE      999.000         1.000
 SITN           0.665       999.000         1.000
 SITN_SE      999.000       999.000       999.000         1.000
```

Positive factor score SE = 0.472
Negative factor score SE = 0.418

Although the variance of each factor was supposed to be 1.0, the variance of the factor scores is < 1.0 because of shrinkage (positive factor var = 0.777; negative factor var = 0.825).

Likewise, the correlation between the factors was .56, but the correlation between the estimated factor scores is .67 instead (given the shrinkage).



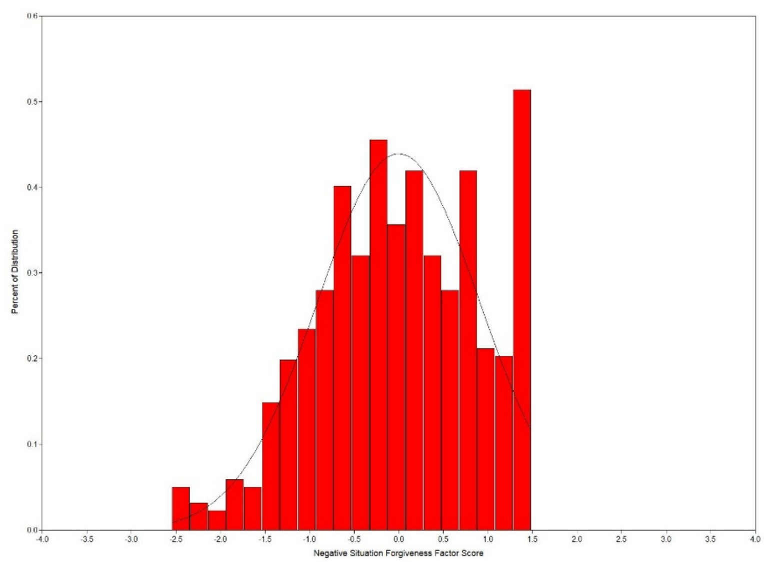
The **positive factor scores** have an estimated mean of 0 with a variance of 0.78 instead of the model-based 1.00.

The SE for each person's factor score is 0.472. Treating factor scores as observed variables is like saying SE = 0.

Positive factor score =
Score ± 2*0.472 = Score ± 0.944!

Positive items factor score reliability =

$$\frac{\sigma_F^2}{\sigma_F^2 + SE_{FS}^2} = \frac{1}{1+0.472^2} = .818$$



The negative factor scores have an estimated mean of 0 with a variance of 0.825 instead of the model-based 1.00.

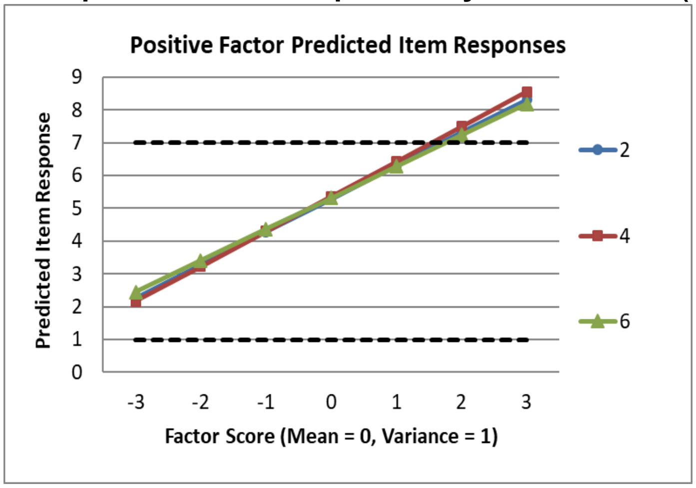
The SE for each person's factor score is 0.418, so ± 0.836!

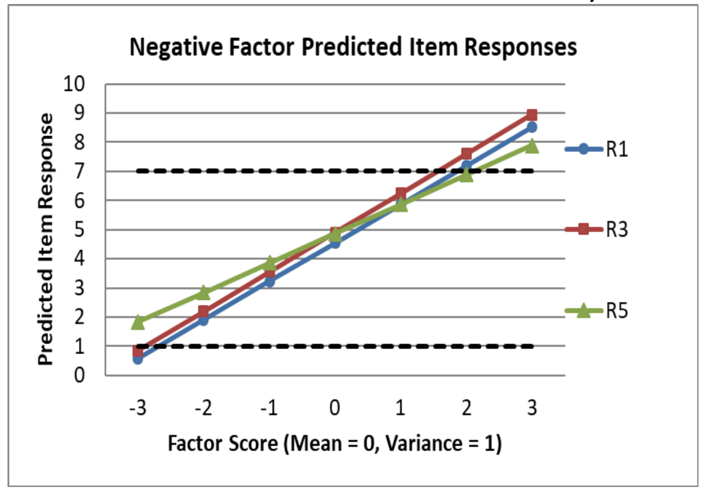
Negative items factor score reliability =

$$\frac{\sigma_F^2}{\sigma_F^2 + SE_{FS}^2} = \frac{1}{1+0.418^2} = .851$$

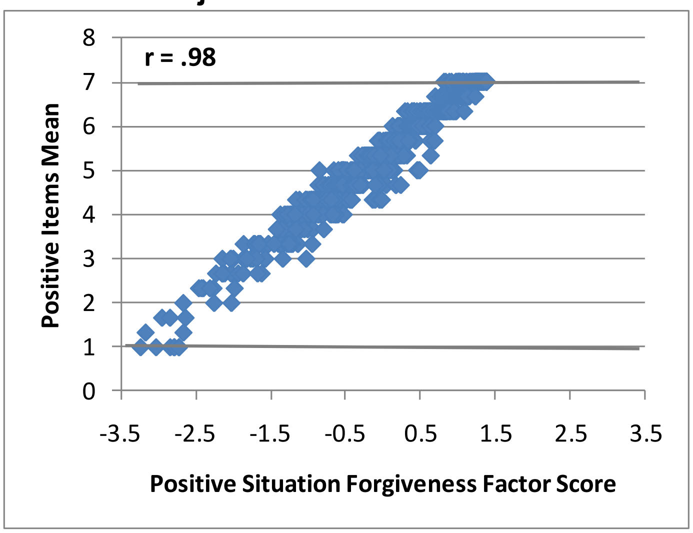
The negative factor scores retain more variance (and have a smaller SE) because there is more information in them, due to higher factor loadings (greater reliability) of their items.

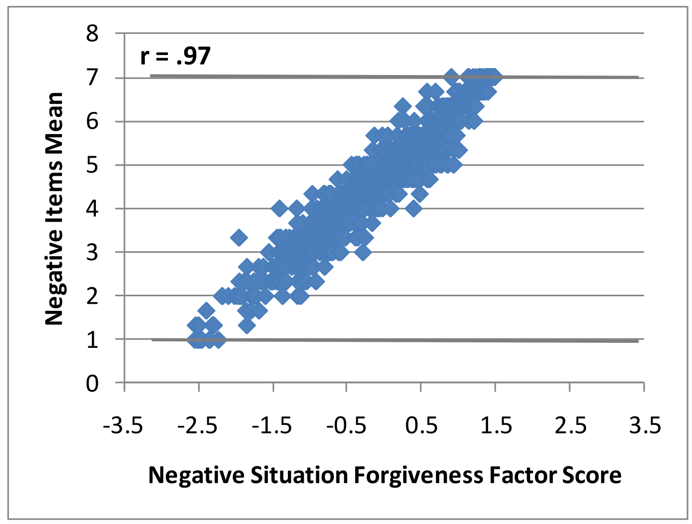
Model-predicted item responses by factor scores (see excel sheet "Factor Model Predictions"):

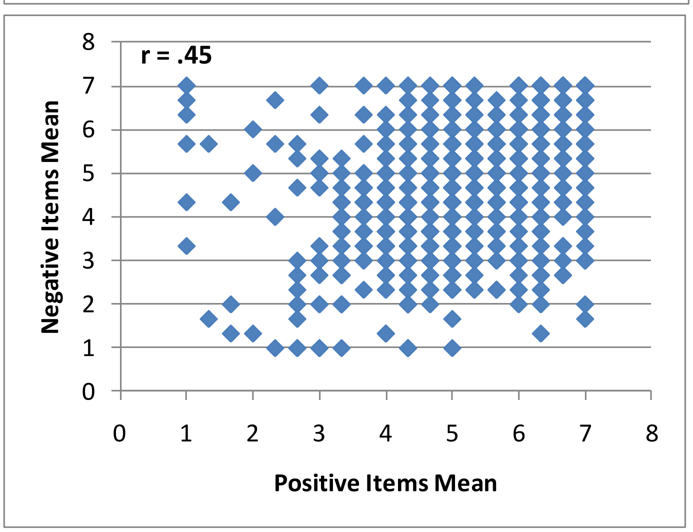


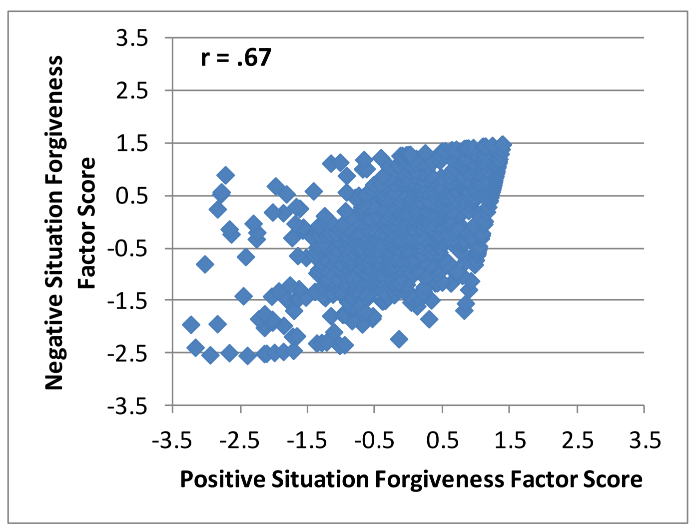


What if we had just taken the mean of the three items for each subscale?









There are problems with either of these observed variable approaches: The **mean of the items** appears to have less variability (i.e., fewer possible scores), and any sum or mean across items assumes that all items should be weighted equally and have no error. The **estimated factor scores** do not have the same properties as estimated for the factor in the model (i.e., less variance for each factor, higher correlation among the factors). **So use SEM if possible!**

Another Example: Formal Tests of CTT Assumptions

We will test the CTT assumption of tau-equivalence (equal factor loadings), one factor at a time.
If those hold, we can then test the assumption of parallel items (equal error variances, too).

First, in Model 5 we will test "tau-equivalence" (equal loadings) of the negative factor only:

```
! Mplus Model 5: Two Factors with Tau-Equivalent Negative Items Only
    SitP BY Sit2*  Sit4  Sit6;                ! SitP loadings (all free)
    SitN BY Sit1r* Sit3r Sit5r (NegLoad);     ! SitN loadings (all held equal)
    SitP@1; SitN@1;                           ! Factor variances (fixed=1)
    SitP WITH SitN*;                          ! Factor covariance (free)
```

```
# Model 5: Two Factors with Tau-Equivalent Negative Items Only (loadings equal)
Syntax5 = "  ! Note labels are in order of inclusion, not according to item name
# Define factor and request item factor loadings all estimated
  SitP =~ L1*Sit2 + L2*Sit4 + L3*Sit6                     # Pos items
  SitN =~ NegLoad*Sit1r + NegLoad*Sit3r + NegLoad*Sit5r    # Neg items
"
# Use MLR estimation like in Mplus, z-score latent factors (mean=0, SD=1)
Model5 = sem(model=Syntax5, data=Example4, estimator="MLR", mimic="mplus", std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution
summary(object=Model5, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

print("LRT: Does the shared loading for the negative items make fit worse?")
anova(Model5, Model4)
```

R lavaan output:

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP =~
    Sit2      (L1)    1.007    0.052   19.491    0.000    1.007    0.730
    Sit4      (L2)    1.063    0.050   21.201    0.000    1.063    0.758
    Sit6      (L3)    0.957    0.052   18.258    0.000    0.957    0.626
  SitN =~
    Sit1r   (NgLd)    1.254    0.032   38.957    0.000    1.254    0.735
    Sit3r   (NgLd)    1.254    0.032   38.957    0.000    1.254    0.805
    Sit5r   (NgLd)    1.254    0.032   38.957    0.000    1.254    0.682

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP ~~
    SitN              0.575    0.041   13.856    0.000    0.575    0.575
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2              5.289    0.042  127.346    0.000    5.289    3.834
   .Sit4              5.359    0.042  126.896    0.000    5.359    3.821
   .Sit6              5.321    0.046  115.492    0.000    5.321    3.477
   .Sit1r             4.547    0.053   86.474    0.000    4.547    2.667
   .Sit3r             4.896    0.048  101.959    0.000    4.896    3.141
   .Sit5r             4.860    0.052   94.060    0.000    4.860    2.644
    SitP              0.000                               0.000    0.000
    SitN              0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2              0.889    0.096    9.216    0.000    0.889    0.467
   .Sit4              0.837    0.092    9.046    0.000    0.837    0.425
   .Sit6              1.425    0.134   10.630    0.000    1.425    0.609
   .Sit1r             1.335    0.083   16.149    0.000    1.335    0.459
   .Sit3r             0.857    0.069   12.337    0.000    0.857    0.353
   .Sit5r             1.806    0.115   15.715    0.000    1.806    0.535
    SitP              1.000                               1.000    1.000
    SitN              1.000                               1.000    1.000
```

Why are the standardized factor loadings for the negative factor not held equal like the unstandardized loadings are?

Mplus Model Fit:

<table>
<tr>
<td>

Fit of previous 2-factor model as baseline:

```
Number of Free Parameters                         19

Loglikelihood
    H0 Value                               -11340.140
    H0 Scaling Correction Factor              1.4017
         for MLR
    H1 Value                               -11322.435
    H1 Scaling Correction Factor              1.4073
         for MLR

RMSEA (Root Mean Square Error Of
Approximation)
    Estimate                                   0.044
    90 Percent C.I.                  0.025     0.064
    Probability RMSEA <= .05                   0.667

CFI/TLI
     CFI                                       0.985
     TLI                                       0.972
```

</td>
<td>

Fit of tau-equivalent negative items 2-factor model:

```
Number of Free Parameters                         17

Loglikelihood
     H0 Value                              -11357.612
     H0 Scaling Correction Factor              1.4474
            for MLR
     H1 Value                              -11322.435
     H1 Scaling Correction Factor              1.4073
            for MLR

RMSEA (Root Mean Square Error Of
Approximation)
      Estimate                                  0.062
      90 Percent C.I.                   0.046   0.079
      Probability RMSEA <= .05                  0.102

CFI/TLI
     CFI                                        0.962
     TLI                                        0.943
```

</td>
</tr>
</table>

Does the assumption of tau-equivalence hold for the negative items? Let’s see the lavaan anova output:

```
[1] "LRT: Does the shared loading for the negative items make fit worse?"
> anova(Model5, Model4)
Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
lavaan NOTE:
    The "Chisq" column contains standard test statistics, not the
    robust test that should be reported per model. A robust difference
    test is a function of two standard (not robust) statistics.

       Df     AIC     BIC   Chisq Chisq diff Df diff     Pr(>Chisq)
Model4  8 22718.3 22813.4 35.4104
Model5 10 22749.2 22834.3 70.3529    34.5013       2 0.000000032221
```

Second, Model 6 tests tau-equivalence of the factor loadings for the positive factor only:

```
! Mplus Model 6: Two Factors with Tau-Equivalent Positive Items Only
    SitP BY Sit2*  Sit4  Sit6 (PosLoad);      ! SitP loadings (all held equal)
    SitN BY Sit1r* Sit3r Sit5r;               ! SitN loadings (all free)
    Sit2  Sit4  Sit6          (E1-E3);        ! SitP residual variances (all free)
    Sit1r Sit3r Sit5r;                        ! SitN residual variances (all free)
    SitP@1; SitN@1;                           ! Factor variances (fixed=1)
    SitP WITH SitN*;                          ! Factor covariance (free)

MODEL CONSTRAINT:
    NEW(AlphaP);   ! This is now equivalent to alpha reliability for pos items
    AlphaP = (1*(PosLoad*3)**2) / ((1*(PosLoad*3)**2) + (E1+E2+E3));
```

```
# Model 6: Two Factors with Tau-Equivalent Positive Items Only (loadings equal)
Syntax6 = "  ! Note labels are in order of inclusion, not according to item name
# Define factor and request item factor loadings all estimated
  SitP =~ PosLoad*Sit2 + PosLoad*Sit4 + PosLoad*Sit6      # Pos items
  SitN =~ L4*Sit1r + L5*Sit3r + L6*Sit5r                  # Neg items
# Item error variances all estimated
  Sit2  ~~ E1*Sit2;  Sit4  ~~ E2*Sit4;  Sit6  ~~ E3*Sit6   # Pos items
  Sit1r ~~ E4*Sit1r; Sit3r ~~ E5*Sit3r; Sit5r ~~ E6*Sit5r  # Neg items
# Calculate Alpha Reliability for Sum Score of Positive Factor:
  AlphaP := ((PosLoad*3)^2) / ( ((PosLoad*3)^2) + (E1 + E2 + E3))
"
# Use MLR estimation like in Mplus, z-score latent factors (mean=0, SD=1)
Model6 = sem(model=Syntax6, data=Example4, estimator="MLR", mimic="mplus", std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution
summary(object=Model6, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

print("LRT: Does the shared loading for the positive items make fit worse?")
anova(Model6, Model4)
```

R lavaan output:

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP =~
    Sit2    (PsLd)    1.014    0.036   28.388    0.000    1.014    0.734
    Sit4    (PsLd)    1.014    0.036   28.388    0.000    1.014    0.733
    Sit6    (PsLd)    1.014    0.036   28.388    0.000    1.014    0.653
  SitN =~
    Sit1r     (L4)    1.325    0.048   27.726    0.000    1.325    0.759
    Sit3r     (L5)    1.349    0.044   30.531    0.000    1.349    0.846
    Sit5r     (L6)    1.010    0.055   18.369    0.000    1.010    0.588

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP ~~
    SitN              0.567    0.040   14.130    0.000    0.567    0.567

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2              5.289    0.042  127.346    0.000    5.289    3.828
   .Sit4              5.359    0.042  126.896    0.000    5.359    3.873
   .Sit6              5.321    0.046  115.492    0.000    5.321    3.426
   .Sit1r             4.547    0.053   86.474    0.000    4.547    2.604
   .Sit3r             4.896    0.048  101.959    0.000    4.896    3.070
   .Sit5r             4.860    0.052   94.060    0.000    4.860    2.832
    SitP              0.000                               0.000    0.000
    SitN              0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2      (E1)    0.881    0.083   10.587    0.000    0.881    0.461
   .Sit4      (E2)    0.886    0.075   11.767    0.000    0.886    0.463
   .Sit6      (E3)    1.384    0.118   11.737    0.000    1.384    0.574
   .Sit1r     (E4)    1.295    0.103   12.580    0.000    1.295    0.425
   .Sit3r     (E5)    0.725    0.092    7.873    0.000    0.725    0.285
   .Sit5r     (E6)    1.925    0.119   16.117    0.000    1.925    0.654
    SitP              1.000                               1.000    1.000
    SitN              1.000                               1.000    1.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    AlphaP            0.746    0.020   38.200    0.000    0.746    0.764

[1] "LRT: Does the shared loading for the positive items make fit worse?"
> anova(Model6, Model4)
Scaled Chi-Squared Difference Test (method = “satorra.bentler.2001”)
lavaan NOTE:
    The “Chisq” column contains standard test statistics, not the
    robust test that should be reported per model. A robust difference
    test is a function of two standard (not robust) statistics.

       Df     AIC     BIC   Chisq Chisq diff Df diff Pr(>Chisq)
Model4  8 22718.3 22813.4 35.4104
Model6 10 22717.5 22802.6 38.6755    2.59719       2    0.27292
```

Mplus model fit:

```
Number of Free Parameters                   17
Loglikelihood
    H0 Value                        -11341.773
    H0 Scaling Correction Factor        1.4187
          for MLR
    H1 Value                        -11322.435
    H1 Scaling Correction Factor        1.4073
          for MLR
```

Does the assumption of tau-equivalence hold for the positive items? How do we know?

Given that tau-equivalence held for the positive factor, in Model 7 we can also test the assumption of parallel items as equal residual variances (in addition to equal factor loadings):

```
! Mplus Model 7: Two Factors with Parallel Positive Items Only
    SitP BY Sit2*  Sit4  Sit6  (PosLoad);   ! SitP loadings (all held equal)
    SitN BY Sit1r* Sit3r Sit5r;             ! SitN loadings (all free)
    Sit2  Sit4  Sit6          (PosError);   ! SitP residual variances (all held equal)
    Sit1r Sit3r Sit5r;                      ! SitN residual variances (all free)
    SitP@1; SitN@1;                         ! Factor variances (fixed=1)
    SitP WITH SitN*;                        ! Factor covariance (free)

MODEL CONSTRAINT:
    NEW(SpearP);   ! This is now equivalent to Spearman-Brown reliability for pos items
    SpearP = (1*(PosLoad*3)**2) / ((1*(PosLoad*3)**2) + (PosError*3));
```

```
# R Model 7: Two Factors with Parallel Positive Items Only (loadings and error variances equal)
Syntax7 = "  ! Note labels are in order of inclusion, not according to item name
# Define factor and request item factor loadings all estimated
  SitP =~ PosLoad*Sit2  + PosLoad*Sit4  + PosLoad*Sit6     # Pos items
  SitN =~ L4*Sit1r + L5*Sit3r + L6*Sit5r                    # Neg items
# Item error variances all estimated
  Sit2  ~~ PosError*Sit2; Sit4  ~~ PosError*Sit4; Sit6  ~~ PosError*Sit6   # Pos items
  Sit1r ~~ E4*Sit1r; Sit3r ~~ E5*Sit3r; Sit5r ~~ E6*Sit5r                  # Neg items
# Calculate Spearman-Brown Reliability for Sum Score of Positive Factor:
  SpearP := ((PosLoad*3)^2) / ( ((PosLoad*3)^2) + (PosError*3)
"
# Use MLR estimation like in Mplus, z-score latent factors (mean=0, SD=1)
Model7 = sem(model=Syntax7, data=Example4, estimator="MLR", mimic="mplus", std.lv=TRUE)
# Print solution: get fit, get effect size, STDYX solution
summary(object=Model7, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

print("LRT: Does the shared error variance for the positive items make fit worse?")
anova(Model7, Model6)
```

R lavaan output—notice that the positive standardized loadings are now equal, too:

```
Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP =~
    Sit2    (PsLd)    1.005    0.035   28.455    0.000    1.005    0.698
    Sit4    (PsLd)    1.005    0.035   28.455    0.000    1.005    0.698
    Sit6    (PsLd)    1.005    0.035   28.455    0.000    1.005    0.698
  SitN =~
    Sit1r     (L4)    1.325    0.048   27.817    0.000    1.325    0.759
    Sit3r     (L5)    1.347    0.044   30.622    0.000    1.347    0.845
    Sit5r     (L6)    1.011    0.055   18.407    0.000    1.011    0.589

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  SitP ~~
    SitN              0.581    0.040   14.582    0.000    0.581    0.581

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2              5.289    0.042  127.346    0.000    5.289    3.676
   .Sit4              5.359    0.042  126.896    0.000    5.359    3.724
   .Sit6              5.321    0.046  115.492    0.000    5.321    3.698
   .Sit1r             4.547    0.053   86.474    0.000    4.547    2.604
   .Sit3r             4.896    0.048  101.959    0.000    4.896    3.070
   .Sit5r             4.860    0.052   94.060    0.000    4.860    2.832
    SitP              0.000                               0.000    0.000
    SitN              0.000                               0.000    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .Sit2    (PsEr)    1.060    0.061   17.452    0.000    1.060    0.512
   .Sit4    (PsEr)    1.060    0.061   17.452    0.000    1.060    0.512
   .Sit6    (PsEr)    1.060    0.061   17.452    0.000    1.060    0.512
   .Sit1r     (E4)    1.294    0.102   12.645    0.000    1.294    0.424
   .Sit3r     (E5)    0.728    0.091    7.992    0.000    0.728    0.286
```

```
   .Sit5r     (E6)    1.922    0.119   16.095    0.000    1.922    0.653
    SitP              1.000                               1.000    1.000
    SitN              1.000                               1.000    1.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    SpearP            0.741    0.020   36.909    0.000    0.741    0.741

[1] "LRT: Does the shared error variance for the positive items make fit worse?"
> anova(Model7, Model6)
Scaled Chi-Squared Difference Test (method = "satorra.bentler.2001")
lavaan NOTE:
    The "Chisq" column contains standard test statistics, not the
    robust test that should be reported per model. A robust difference
    test is a function of two standard (not robust) statistics.

      Df     AIC     BIC   Chisq Chisq diff Df diff  Pr(>Chisq)
Model6 10 22717.5 22802.6 38.6755
Model7 12 22753.9 22829.0 79.0488    20.4253       2 0.000036703
```

Mplus model fit:

```
Number of Free Parameters                        15

Loglikelihood
          H0 Value                       -11361.960
          H0 Scaling Correction Factor       1.3443
            for MLR
          H1 Value                       -11322.435
          H1 Scaling Correction Factor       1.4073
            for MLR
```

Does the assumption of parallel items hold for the positive items? How do we know?

Example results section describing these analyses…

(Note: You may borrow the phrasing contained in this example to describe various aspects of your homework analyses, but your own results sections will not mimic this example exactly—they should be customized to describe the what and the why of what you did, specifically, including theoretical motivation for any model modifications).

(Descriptive information for the sample and items would have already been given in the method section…)

The reliability and dimensionality of six items assessing forgiveness of situations were evaluated in a sample of 1,103 persons with confirmatory factor analyses using robust maximum likelihood estimation (MLR) in M*plus* v. 8.10 (Muthén & Muthén, 1998–2017) and the package lavaan v. 0.6-15 (Rossell, 2012) within R v. 4.3.2. All models were identified by setting any latent factor means to 0 and latent factor variances to 1, such that all item intercepts, item factor loadings, and item residual variances were estimated. The six items utilized a seven-point response scale, and three items were reverse-coded prior to analysis such that higher values then indicated greater levels of situation forgiveness for all items. As reported in Table 1, model fit statistics include the obtained model χ^2, its scaling factor (in which values different than 1.000 indicate deviations from multivariate normality), its degrees of freedom, and its *p*-value (in which non-significance is traditionally desirable for good fit), CFI (the Comparative Fit Index, in which values higher than .95 are traditionally desirable for good fit), and the RMSEA (the Root Mean Square Error of Approximation point estimate and 90% confidence interval, in which values lower than .06 are traditionally desirable for good fit). As reported in Table 2, nested model comparisons were conducted using the rescaled $-2\Delta LL$ with degrees of freedom equal to the rescaled difference in the number of parameters between models (i.e., a rescaled likelihood ratio test). The specific models examined are described below.

Although a one-factor model was initially posited to account for the pattern of covariance across these six items, it resulted in poor fit, as shown in Table 1. Although each item had a significant factor loading (with standardized loadings ranging from .509 to .778), a single latent factor did not adequately describe the pattern of relationship across these six items as initially hypothesized. Sources of local misfit were identified using the normalized residual covariance matrix, in which individual values were calculated as: (observed covariance – expected covariance) / SE of observed covariance. Relatively large positive residual covariances were observed among items 2, 4, and 6 (the positively-worded items), indicating that these items were more related than was predicted by the single-factor model. Modification indices corroborated this pattern, further suggesting additional remaining relationships among the negatively-worded items as well.

The necessity of separate latent factors for the positively-worded and negatively-worded items was tested by specifying a two-factor model in which the positively-worded items 2, 4, and 6 indicated a *forgiveness* factor, and in which negatively-worded items 1, 3, and 5 indicated a *not unforgiveness* factor, and in which the two factors were allowed to covary. The two-factor model fit was acceptable by every criterion except the significant χ^2 (which was likely due to the large sample). In addition, the two-factor model fit significantly better than the one-factor model, as reported in Table 2, indicating that the estimated correlation between the two factors of .564 was significantly less than 1 (which would be consistent with a single factor instead). Thus, the six items appeared to measure two separate but related constructs. Further examination of local fit via normalized residual covariances and modification indices yielded no interpretable remaining relationships, and thus this two-factor model was retained.

Table 3 provides the estimates and their standard errors for the item factor loadings, intercepts, and residual (error) variances from both the unstandardized and standardized solutions. All factor loadings and the factor covariance were statistically significant. As shown in Table 3, standardized loadings for the forgiveness factor items ranged from .625 to .759 (with R^2 values for the amount of item variance attributable to the factor ranging from .390 to .575), and standardized loadings for the not unforgiveness factor ranged from .588 to .846 (with R^2 values of .346 to .715), suggesting the factor loadings were practically significant as well. Omega model-based reliability was calculated for the sum scores of each factor as described in Brown (2015) as the squared sum of the factor loadings divided by the squared sum of the factor loadings plus the sum of the error variances plus twice the sum of the error covariances (although no error covariances were included here). Omega was .744 for the forgiveness factor and .775 for the not unforgiveness factor, suggesting marginal reliability for each of the three-item sum scores.

The resulting distribution of the factors was examined by requesting empirical Bayes estimates of the individual scores for each factor, as shown in Figure 1. Factor score standard errors for the forgiveness and not unforgiveness factors were.472 and .418, respectively. Factor score reliability was computed as the factor variance (fixed to 1) divided by that plus the squared factor score standard errors. Factor score reliability was .818 for the forgiveness factor and .851 for the not unforgiveness factor, suggesting acceptable reliability for each of the three-item factor scores.

In addition, Figure 2 shows the predicted response for each item as a linear function of the latent factor based on the estimated model parameters. As shown, the predicted item response goes above the highest response option just before a latent factor score of +2 (i.e., 2 SDs above the mean), resulting in a ceiling effect for both sets of factor scores, as also shown in Figure 1. In addition, for the not unforgiveness factor, the predicted item response goes below the lowest response option just before a latent factor score of −3 (i.e., 3 SDs below the mean), resulting in a floor effect for the not unforgiveness factor, as also shown in Figure 1.

The extent to which the items within each factor could be seen as exchangeable was then examined via an additional set of nested model comparisons, as reported in Table 1 (for fit) and Table 2 (for comparisons of fit). First, the assumption of tau-equivalence (i.e., true-score equivalence, equal discrimination across items) was examined by constraining the factor loadings to be equal within a factor. For the not unforgiveness factor, the tau-equivalent model fit was acceptable but was significantly worse than the original two-factor model fit (i.e., in which all loadings were estimated freely). For the forgiveness factor, however, the fit of the tau-equivalent model was acceptable and was not significantly worse than the fit of the original two-factor model. Thus, the assumption of tau-equivalence held for the forgiveness factor items only. Finally, the assumption of parallel items (i.e., equal factor loadings and equal residual variances, or equal reliability across items) was examined for the forgiveness factor items only, and the resulting model fit was acceptable but was significantly worse than the fit of the tau-equivalent forgiveness factor model. Thus, the assumption of parallel items did not hold for the forgiveness factor items. In summary, while the not unforgiveness factor items were not exchangeable, the forgiveness factor items appeared exchangeable with respect to their factor loadings only (i.e., equal discrimination, but not equal item residual variances or item reliability).

Tables would be built as seen in the excel workbook:

Table 1 → “Model Fit Table 1” worksheet
Table 2 → “MLR Comparisons Table 2” worksheet
Table 3 → “Model Estimates Table 3” worksheet

Figures would be built as seen in this example:

Figure 1 → Can be built in Mplus or ggplot2 in R
Figure 2 → Can be built using “Factor Model Predictions” worksheet

References:

Brown, T. A. (2015). *Confirmatory factor analysis for applied research* (2nd ed.). New York, NY: Guilford.

Muthén, L. K., & Muthén, B.O. (1998–2017). *Mplus user's guide* (8th ed.). Los Angeles, CA: Muthén & Muthén.

R Core Team (2017). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rosseel, Y (2012). lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2), 1-36. URL http://www.jstatsoft.org/v48/i02/.

Example 4 Continued: CFA of Forgiveness of Situations (N = 1103) using SAS MIXED

SAS Code to Read in Mplus Data:

```
* Import data from Mplus, becomes var1-var23 without names at top;
PROC IMPORT OUT=work.Situation DATAFILE= "&example.\Example4_Data.csv" DBMS=CSV REPLACE;
     GETNAMES=NO; DATAROW=1; RUN;

* Rename variables, remove missing values;
DATA work.Situation; SET work.Situation;
     ARRAY old(23) var1-var23;
     ARRAY new(23) PersonID Self1 Self2r Self3 Self4r Self5 Self6r
                   Other1r Other2 Other3r Other4 Other5r Other6
                   Sit1r Sit2 Sit3r Sit4 Sit5r Sit6
                   Selfsub Othsub Sitsub HFSsum;
       DO i=1 TO 23; new(i)=old(i); IF new(i)=99999 THEN new(i)=.; END;
       DROP i var1-var23; RUN;

* Stack situation items into one column, as required by MIXED;
DATA work.SituationStacked; SET work.Situation;
       ARRAY aitem(6)  Sit1r Sit2 Sit3r Sit4 Sit5r Sit6;
       DO i=1 TO 6; itemnum=i; response=aitem(i); OUTPUT; END; DROP i; RUN;
```

Independence (Null) Baseline Model: Item means and variances, but NO covariances

```
TITLE "Independence (Null) CFA Model in MIXED";
PROC MIXED DATA=work.SituationStacked NOCLPRINT COVTEST IC NAMELEN=100 METHOD=ML;
       CLASS PersonID itemnum;
       MODEL response = itemnum / SOLUTION NOINT NOTEST;
       REPEATED itemnum / TYPE=TOEPH(1) SUBJECT=PersonID R; RUN;
```

TYPE=TOEPH(1) predicts a diagonal matrix that would be the same as TYPE=UN(1).

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	3.0493					
2		1.9028				
3			2.5431			
4				1.9672		
5					2.9451	
6						2.3412

The **R** matrix shows the unconditional variances per item—repeated in the next piece of output as Var(item). Note that this independence "null" model predicts NO covariances between items.

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
Var(1)	PersonID	3.0493	0.1298	23.48	<.0001
Var(2)	PersonID	1.9028	0.08102	23.48	<.0001
Var(3)	PersonID	2.5431	0.1083	23.48	<.0001
Var(4)	PersonID	1.9672	0.08377	23.48	<.0001
Var(5)	PersonID	2.9451	0.1254	23.48	<.0001
Var(6)	PersonID	2.3412	0.09969	23.48	<.0001

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
24625.9	12	24649.9	24650.0	24672.6	24710.0	24722.0

Model fit is given as −2LL rather than LL (but otherwise is the same as given from Mplus).

Solution for Fixed Effects

Effect	itemnum	Estimate	Standard Error	DF	t Value	Pr > |t|
itemnum	1	4.5467	0.05258	5509	86.47	<.0001
itemnum	2	5.2892	0.04153	5509	127.35	<.0001
itemnum	3	4.8957	0.04802	5509	101.96	<.0001
itemnum	4	5.3590	0.04223	5509	126.90	<.0001
itemnum	5	4.8604	0.05167	5509	94.06	<.0001
itemnum	6	5.3209	0.04607	5509	115.49	<.0001

The fixed effects show the unconditional means per item.

Saturated (Unstructured) Baseline Model: Item means, variances, and covariances in original data

```
TITLE "Saturated (Unstructured) CFA Model in MIXED";
```

```
PROC MIXED DATA=work.SituationStacked NOCLPRINT COVTEST IC NAMELEN=100 METHOD=ML;
       CLASS PersonID itemnum;
       MODEL response = itemnum / SOLUTION NOINT NOTEST;
       REPEATED itemnum / TYPE=UN(6) SUBJECT=PersonID R RCORR; RUN;
```

TYPE=UN(6) predicts a fully-estimated **R** matrix with no constraints whatsoever.

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	3.0493	0.5772	1.8022	0.7339	1.3583	0.7946
2	0.5772	1.9028	0.6974	1.1029	0.6043	0.9652
3	1.8022	0.6974	2.5431	0.8244	1.3191	0.8676
4	0.7339	1.1029	0.8244	1.9672	0.6947	0.9618
5	1.3583	0.6043	1.3191	0.6947	2.9451	0.7982
6	0.7946	0.9652	0.8676	0.9618	0.7982	2.3412

Estimated R Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.2396	0.6472	0.2997	0.4533	0.2974
2	0.2396	1.0000	0.3170	0.5700	0.2553	0.4573
3	0.6472	0.3170	1.0000	0.3686	0.4820	0.3555
4	0.2997	0.5700	0.3686	1.0000	0.2886	0.4482
5	0.4533	0.2553	0.4820	0.2886	1.0000	0.3040
6	0.2974	0.4573	0.3555	0.4482	0.3040	1.0000

The **R** matrix shows the unconditional variances and covariances for the items.

RCORR is the unconditional correlation matrix.

Note THIS IS THE DATA—the only discrepancies you'd see relative to descriptive statistics would be from missing data, as these are ML estimates (that assume MAR rather than MCAR as in listwise deletion).

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	PersonID	3.0493	0.1298	23.48	<.0001
UN(2,1)	PersonID	0.5772	0.07458	7.74	<.0001
UN(2,2)	PersonID	1.9028	0.08102	23.48	<.0001
UN(3,1)	PersonID	1.8022	0.09988	18.04	<.0001
UN(3,2)	PersonID	0.6974	0.06948	10.04	<.0001
UN(3,3)	PersonID	2.5431	0.1083	23.48	<.0001
UN(4,1)	PersonID	0.7339	0.07699	9.53	<.0001
UN(4,2)	PersonID	1.1029	0.06705	16.45	<.0001
UN(4,3)	PersonID	0.8244	0.07178	11.49	<.0001
UN(4,4)	PersonID	1.9672	0.08377	23.48	<.0001
UN(5,1)	PersonID	1.3583	0.09907	13.71	<.0001
UN(5,2)	PersonID	0.6043	0.07356	8.21	<.0001
UN(5,3)	PersonID	1.3191	0.09148	14.42	<.0001
UN(5,4)	PersonID	0.6947	0.07543	9.21	<.0001
UN(5,5)	PersonID	2.9451	0.1254	23.48	<.0001
UN(6,1)	PersonID	0.7946	0.08393	9.47	<.0001
UN(6,2)	PersonID	0.9652	0.06988	13.81	<.0001
UN(6,3)	PersonID	0.8676	0.07798	11.13	<.0001
UN(6,4)	PersonID	0.9618	0.07081	13.58	<.0001
UN(6,5)	PersonID	0.7982	0.08264	9.66	<.0001
UN(6,6)	PersonID	2.3412	0.09969	23.48	<.0001

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
22644.9	27	22698.9	22699.1	22750.0	22834.0	22861.0

Solution for Fixed Effects

Effect	itemnum	Estimate	Standard Error	DF	t Value	Pr > |t|
itemnum	1	4.5467	0.05258	5509	86.47	<.0001
itemnum	2	5.2892	0.04153	5509	127.35	<.0001
itemnum	3	4.8957	0.04802	5509	101.96	<.0001
itemnum	4	5.3590	0.04223	5509	126.90	<.0001
itemnum	5	4.8604	0.05167	5509	94.06	<.0001
itemnum	6	5.3209	0.04607	5509	115.49	<.0001

The fixed effects again show the unconditional means per item.

Model 1. Single Factor with Fully Z-Scored Factor Model Identification (Factor Variance = 1, Factor Mean = 0, All Loadings and Intercepts Estimated)

```
TITLE "Single-Factor CFA Model (Factor Variance=1, Factor Mean=0) in MIXED";
PROC MIXED DATA=work.SituationStacked NOCLPRINT COVTEST IC NAMELEN=100 METHOD=ML;
       CLASS PersonID itemnum;
       MODEL response = itemnum / SOLUTION NOINT NOTEST;
       REPEATED itemnum / TYPE=FA(1) SUBJECT=PersonID R RCORR;
RUN;
```

TYPE=FA(1) creates the covariance matrix that would be predicted by a single-factor CFA model.

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	3.0493	0.8670	1.5313	0.9682	1.2626	1.0108
2	0.8670	1.9028	0.8716	0.5511	0.7187	0.5753
3	1.5313	0.8716	2.5431	0.9733	1.2692	1.0161
4	0.9682	0.5511	0.9733	1.9672	0.8025	0.6424
5	1.2626	0.7187	1.2692	0.8025	2.9451	0.8378
6	1.0108	0.5753	1.0161	0.6424	0.8378	2.3412

The **R** matrix shows the predicted variances and covariances for the items.

RCORR is the single-factor predicted correlation matrix.

THIS IS NO LONGER THE DATA. So the objective is to see how close this predicted covariance matrix is from the one given by the saturated model (which was the data).

Estimated R Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.3600	0.5499	0.3953	0.4213	0.3783
2	0.3600	1.0000	0.3962	0.2848	0.3036	0.2726
3	0.5499	0.3962	1.0000	0.4351	0.4638	0.4164
4	0.3953	0.2848	0.4351	1.0000	0.3334	0.2994
5	0.4213	0.3036	0.4638	0.3334	1.0000	0.3191
6	0.3783	0.2726	0.4164	0.2994	0.3191	1.0000

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
FA(1)	PersonID	1.5259	0.09440	16.16	<.0001
FA(2)	PersonID	1.4093	0.07096	19.86	<.0001
FA(3)	PersonID	1.0038	0.07755	12.94	<.0001
FA(4)	PersonID	1.3518	0.07071	19.12	<.0001
FA(5)	PersonID	1.8986	0.09312	20.39	<.0001
FA(6)	PersonID	1.6706	0.08330	20.05	<.0001
FA(1,1)	PersonID	1.2342	0.05332	23.15	<.0001
FA(2,1)	PersonID	0.7025	0.04720	14.88	<.0001
FA(3,1)	PersonID	1.2407	0.04783	25.94	<.0001
FA(4,1)	PersonID	0.7845	0.04679	16.76	<.0001
FA(5,1)	PersonID	1.0230	0.05202	19.67	<.0001
FA(6,1)	PersonID	0.8190	0.05019	16.32	<.0001

The FA(item) terms are the item residual variances. The FA(item, factor) terms are the item factor loadings.

So the total variance per item is given by: $\text{loading}^2(1)$ + error variance, as shown in the **R** matrix above.

Item 1 = 1.2342^2(1) + 1.5259 = 3.0493

The covariance between items is given by their loadings multiplied together.

Item 1 and 2 cov = 1.2342*0.7025 = 0.8670

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
23072.8	18	23108.8	23108.9	23142.9	23198.9	23216.9

Solution for Fixed Effects

Effect	itemnum	Estimate	Standard Error	DF	t Value	Pr > |t|
itemnum	1	4.5467	0.05258	5509	86.47	<.0001
itemnum	2	5.2892	0.04153	5509	127.35	<.0001
itemnum	3	4.8957	0.04802	5509	101.96	<.0001
itemnum	4	5.3590	0.04223	5509	126.90	<.0001
itemnum	5	4.8604	0.05167	5509	94.06	<.0001
itemnum	6	5.3209	0.04607	5509	115.49	<.0001

The fixed effects now show the intercepts per item conditional on factor = 0 (which then are equal to the original item means).

Tau-Equivalent Items Single-Factor Model with Marker Item Factor Model Identification
(Factor Variance = ?, Factor Mean = 0, All Loadings Equal at 1)

```
TITLE "Tau-Equivalent Items Single-Factor CFA Model (Factor Variance=1, Factor Mean=0) in MIXED";
PROC MIXED DATA=work.SituationStacked NOCLPRINT COVTEST IC NAMELEN=100 METHOD=ML;
       CLASS PersonID itemnum;
       MODEL response = itemnum / SOLUTION NOINT NOTEST;
       RANDOM INTERCEPT / TYPE=UN SUBJECT=PersonID  G V VCORR;
       REPEATED itemnum / TYPE=TOEPH(1) SUBJECT=PersonID R; RUN;
```

A random intercept creates a constant source of covariance across all items.

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	2.0017					
2		1.1357				
3			1.4550			
4				1.0866		
5					2.0552	
6						1.4565

Estimated G Matrix

Row	Effect	PersonID	Col1
1	Intercept	1	0.9127

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	2.9143	0.9127	0.9127	0.9127	0.9127	0.9127
2	0.9127	2.0483	0.9127	0.9127	0.9127	0.9127
3	0.9127	0.9127	2.3677	0.9127	0.9127	0.9127
4	0.9127	0.9127	0.9127	1.9993	0.9127	0.9127
5	0.9127	0.9127	0.9127	0.9127	2.9679	0.9127
6	0.9127	0.9127	0.9127	0.9127	0.9127	2.3691

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.3735	0.3474	0.3781	0.3103	0.3473
2	0.3735	1.0000	0.4144	0.4510	0.3702	0.4143
3	0.3474	0.4144	1.0000	0.4195	0.3443	0.3853
4	0.3781	0.4510	0.4195	1.0000	0.3747	0.4194
5	0.3103	0.3702	0.3443	0.3747	1.0000	0.3442
6	0.3473	0.4143	0.3853	0.4194	0.3442	1.0000

The **R** matrix shows the item residual variances.

The **G** matrix shows the variance due to the factor for all items.

V is the predicted covariance matrix from putting **G** and **R** back together, and **VCORR** is the predicted correlation matrix.

Aligning notation
In MLM = In CFA:

V = Sigma Σ
Z = Lambda Λ
G = Phi Φ (factor var–cov)
R = Psi Ψ (error var–cov)

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	PersonID	0.9127	0.04938	18.48	<.0001
Var(1)	PersonID	2.0017	0.09613	20.82	<.0001
Var(2)	PersonID	1.1357	0.05929	19.15	<.0001
Var(3)	PersonID	1.4550	0.07304	19.92	<.0001
Var(4)	PersonID	1.0866	0.05703	19.05	<.0001
Var(5)	PersonID	2.0552	0.09729	21.13	<.0001
Var(6)	PersonID	1.4565	0.07161	20.34	<.0001

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
23131.1	13	23157.1	23157.1	23181.7	23222.2	23235.2

Solution for Fixed Effects

Effect	itemnum	Estimate	Standard Error	DF	t Value	Pr > |t|
itemnum	1	4.5467	0.05140	5510	88.45	<.0001
itemnum	2	5.2892	0.04309	5510	122.74	<.0001
itemnum	3	4.8957	0.04633	5510	105.67	<.0001
itemnum	4	5.3590	0.04257	5510	125.87	<.0001
itemnum	5	4.8604	0.05187	5510	93.70	<.0001
itemnum	6	5.3209	0.04635	5510	114.81	<.0001

The fixed effects still show the intercepts per item conditional on factor = 0 (which then are equal to the original item means).

Parallel Items Single-Factor Model with Marker Item Factor Model Identification
(Factor Variance = ?, Factor Mean = 0, All Loadings = 1 and All Error Variances Equal)

```
TITLE "Parallel Items Single-Factor CFA Model (Factor Variance=1, Factor Mean=0) in MIXED";
PROC MIXED DATA=work.SituationStacked NOCLPRINT COVTEST IC NAMELEN=100 METHOD=ML;
       CLASS PersonID itemnum;
       MODEL response = itemnum / SOLUTION NOINT NOTEST;
```

A random intercept creates a constant source of covariance across all items. A Type=VC R matrix means equal residual variance across items.

```
RANDOM INTERCEPT / TYPE=UN SUBJECT=PersonID G V VCORR;
REPEATED itemnum / TYPE=VC SUBJECT=PersonID R; RUN;
```

Estimated R Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.5180					
2		1.5180				
3			1.5180			
4				1.5180		
5					1.5180	
6						1.5180

Estimated G Matrix

Row	Effect	Person ID	Col1
1	Intercept	1	0.9401

Estimated V Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	2.4581	0.9401	0.9401	0.9401	0.9401	0.9401
2	0.9401	2.4581	0.9401	0.9401	0.9401	0.9401
3	0.9401	0.9401	2.4581	0.9401	0.9401	0.9401
4	0.9401	0.9401	0.9401	2.4581	0.9401	0.9401
5	0.9401	0.9401	0.9401	0.9401	2.4581	0.9401
6	0.9401	0.9401	0.9401	0.9401	0.9401	2.4581

Estimated V Correlation Matrix for PersonID 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	1.0000	0.3825	0.3825	0.3825	0.3825	0.3825
2	0.3825	1.0000	0.3825	0.3825	0.3825	0.3825
3	0.3825	0.3825	1.0000	0.3825	0.3825	0.3825
4	0.3825	0.3825	0.3825	1.0000	0.3825	0.3825
5	0.3825	0.3825	0.3825	0.3825	1.0000	0.3825
6	0.3825	0.3825	0.3825	0.3825	0.3825	1.0000

The **R** matrix shows the item residual variances.

The **G** matrix shows the variance due to the factor for all items.

V is the predicted covariance matrix from putting **G** and **R** back together, and **VCORR** is the predicted correlation matrix.

This type of predicted marginal **V** covariance matrix has a special name in MLM: **compound symmetry**.

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	PersonID	0.9401	0.05103	18.42	<.0001
itemnum	PersonID	1.5180	0.02891	52.51	<.0001

Information Criteria

Neg2LogLike	Parms	AIC	AICC	HQIC	BIC	CAIC
23254.0	8	23270.0	23270.1	23285.2	23310.1	23318.1

Solution for Fixed Effects

Effect	itemnum	Estimate	Standard Error	DF	t Value	Pr > |t|
itemnum	1	4.5467	0.04721	5510	96.31	<.0001
itemnum	2	5.2892	0.04721	5510	112.04	<.0001
itemnum	3	4.8957	0.04721	5510	103.71	<.0001
itemnum	4	5.3590	0.04721	5510	113.52	<.0001
itemnum	5	4.8604	0.04721	5510	102.96	<.0001
itemnum	6	5.3209	0.04721	5510	112.71	<.0001

The fixed effects still show the intercepts per item conditional on factor = 0 (which then are equal to the original item means).

Unfortunately, multiple factor models in MIXED appear to be EFA models instead of CFA models, so no examples of two-factor models are given here. PROC CALIS can be used for CFA in SAS.