

# CFA Identification & Estimation

## Theory Construction and Statistical Modeling



**Utrecht  
University**

Kyle M. Lang

Department of Methodology & Statistics  
Utrecht University

# Outline

---

## SAPI Data

## Flavors of Latent Construct

- Reflective Constructs

- Formative Constructs

## Model Estimation

- Model-Implied Statistics

## Model Identification

- Degrees of Freedom

- Just-Identified Models

- Under-Identified Models

- Over-Identified Models

- Multiple Factors

## Example



# South African Personality Inventory Project



Carin Hill  
Leon Jackson  
Deon Meiring  
J. Aleweyn Nel

Ian Rothmann  
Michael Temane  
Velichko H. Valchev  
Fons J. R. van de Vijver

Nel, J. A., Valchev, V. H., Rothmann, S., van de Vijver, F. J. R., Meiring, D., & de Bruin, G. P. (2012). Exploring the personality structure in the 11 languages of South Africa. *Journal of Personality*, 80, 915–948.

# SAPI details

---

- 1216 participants from 11 official language groups
- From about 50,000 descriptive responses to 262 personality items
- Nine personality clusters:
  - Conscientiousness
  - Emotional Stability
  - Extraversion
  - Facilitating
  - Integrity
  - Intellect
  - Openness
  - Relationship Harmony
  - Soft-Heartedness (Ubuntu)
- Our data: selection of 1000 participants



# FLAVORS OF LATENT CONSTRUCT

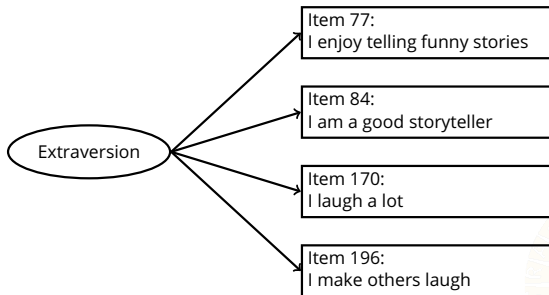


# Reflective Constructs

---

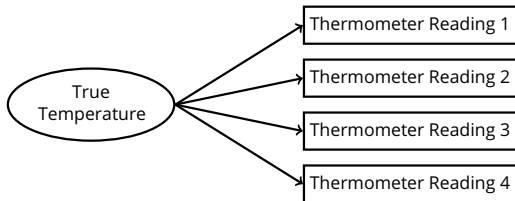
In a reflective measurement model, the items are the dependent variables, and the latent factor is the independent variable.

- The observed items are dependent variables.
- The latent factor is causing the items to take the values we observe.



# Reflective Constructs

---

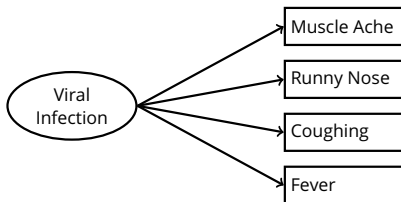


The true temperature is the causal factor that produces thermometer readings.

- Any given thermometer reading is only an imperfect reflection of the true temperature.
- Taking multiple readings increases the reliability of our temperature estimate.

# Reflective Constructs

---



The latent viral infection is the causal factor that gives rise to the observed symptoms of illness.

- Symptoms are the dependent variables.
- Viral infection is the unobserved predictor variable.

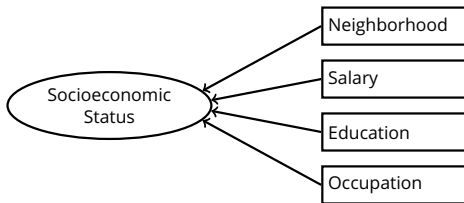




# Formative Constructs

---

In a formative measurement model, the direction of the factor loadings is flipped.



SES is an *index* defined as a (weighted) sum of the observed items.

- SES is the (latent) dependent variable, predicted by the items.
- This model is not empirically testable.

# MODEL ESTIMATION



# Model-Implied Statistics

---

Most statistical estimation algorithms operate by minimizing the difference between two key reference points:

1. The *model-implied* statistics/predictions/fitted values
  - The sufficient statistics implied by the structure of your model.
  - Predicted/fitted values produced by your model.
2. The *observed* statistics/values
  - The sufficient statistics calculated from the observed data.
  - The raw outcome values from your dataset.

The predictions/implied statistics produced by a good model must be simpler than the analagous quantities in the observed data.

- A model that exactly replicates the observed data is overfitting.
- The inferences from such models won't generalize to the population.

# Model-Implied Statistics

---

You should already be familiar with this idea from OLS regression.

- The fitted values are the model implied statistics.

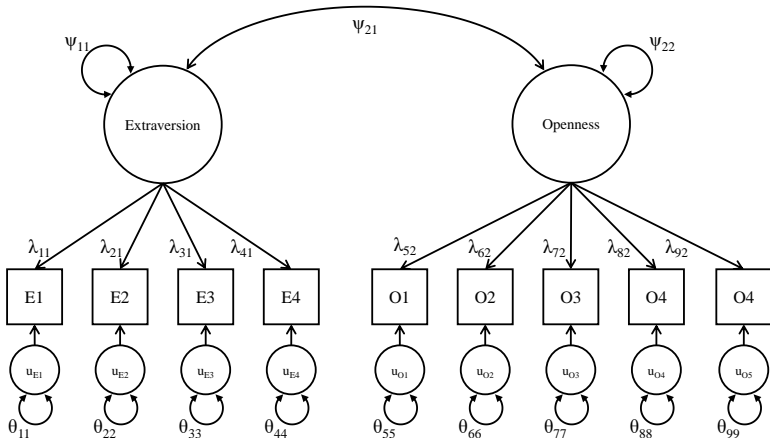
$$\hat{Y}_n = \hat{\beta}_0 + \sum_{p=1}^P \hat{\beta}_p X_{n,p}$$

- The raw outcome variable,  $Y$ , contains the observed values.
- Minimize the difference between  $\hat{Y}$  and  $Y$  to estimate the model.

$$RSS = \sum_{n=1}^N (Y_n - \hat{Y}_n)^2$$



# Fully Specified Path Diagram



# Parameter Matrices

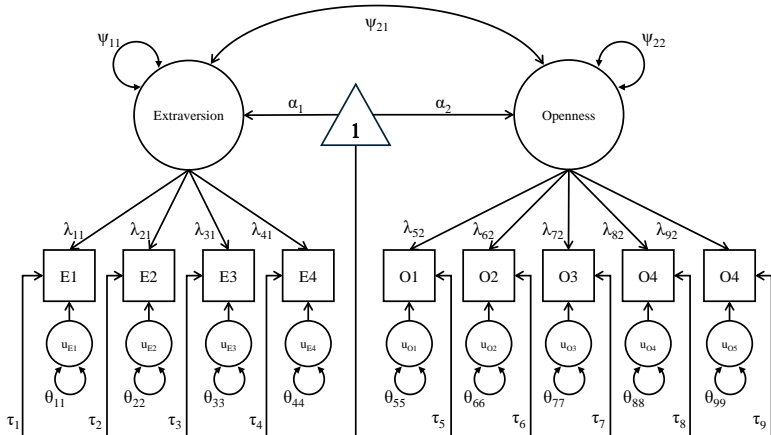
---

$$\Psi = \begin{bmatrix} \psi_{11} & \\ \psi_{21} & \psi_{22} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$

# Fully Specified Path Diagram



# Parameter Matrices

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \Psi = \begin{bmatrix} \psi_{11} & \\ \psi_{21} & \psi_{22} \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \\ \tau_8 \\ \tau_9 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$



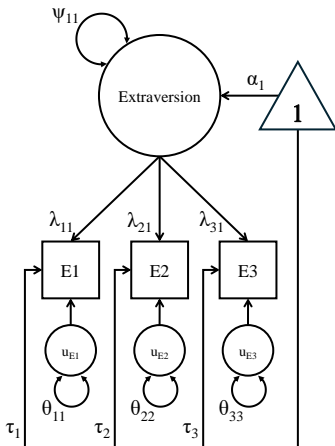
# Parameter Matrices

To see what role these parameter matrices play in model estimation, we'll work with a simpler example.

$$\alpha = [\alpha_1] \quad \Psi = [\psi_{11}]$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & \\ 0 & \theta_{22} & \\ 0 & 0 & \theta_{33} \end{bmatrix}$$



# Model-Implied Statistics

---

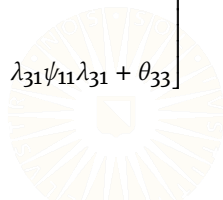
The parameter matrices define the model-implied mean vector,  $\hat{\mu}$ , and covariance matrix,  $\hat{\Sigma}$ , via the following formulas.

$$\Sigma = \Lambda\Psi\Lambda^T + \Theta$$

$$\mu = \tau + \Lambda\alpha$$

By expanding these formulas, we can see how the model reproduces each mean, variance, and covariance.

$$\hat{\mu} = \begin{bmatrix} \tau_1 + \lambda_{11}\alpha_1 \\ \tau_2 + \lambda_{22}\alpha_1 \\ \tau_3 + \lambda_{33}\alpha_1 \end{bmatrix} \quad \hat{\Sigma} = \begin{bmatrix} \lambda_{11}\psi_{11}\lambda_{11} + \theta_{11} & & \\ \lambda_{11}\psi_{11}\lambda_{21} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \\ \lambda_{11}\psi_{11}\lambda_{31} & \lambda_{21}\psi_{11}\lambda_{31} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \end{bmatrix}$$



# Model-Implied Statistics

The estimating algorithm chooses values for  $\alpha$ ,  $\tau$ ,  $\Psi$ , and  $\Lambda$  that minimize the differences between the model-implied statistics,  $\{\hat{\mu}, \hat{\Sigma}\}$ , and the sufficient statistics calculated from the observed data,  $\{\tilde{\mathbf{Y}}, \text{Cov}(\mathbf{Y})\}$ .

$$\hat{\mu} = \begin{bmatrix} \tau_1 + \lambda_{11}\alpha_1 \\ \tau_2 + \lambda_{22}\alpha_1 \\ \tau_3 + \lambda_{33}\alpha_1 \end{bmatrix} \quad \hat{\Sigma} = \begin{bmatrix} \lambda_{11}\psi_{11}\lambda_{11} + \theta_{11} & & \\ \lambda_{11}\psi_{11}\lambda_{21} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \\ \lambda_{11}\psi_{11}\lambda_{31} & \lambda_{21}\psi_{11}\lambda_{31} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \end{bmatrix}$$

$$\tilde{\mathbf{Y}} = \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{bmatrix} \quad \text{Cov}(\mathbf{Y}) = \begin{bmatrix} \text{var}(y_1) & & \\ \text{cov}(y_2, y_1) & \text{var}(y_2) & \\ \text{cov}(y_3, y_1) & \text{cov}(y_3, y_2) & \text{var}(y_3) \end{bmatrix}$$

# Optimization Objective

---

We can formulate the familiar OLS objective as an abstract optimization problem as follows.

$$f(\beta_0, \beta_1, \dots, \beta_p) = \arg \min_{\beta_0, \beta_1, \dots, \beta_p} \|Y - \hat{Y}\|$$

Applying the same idea to our CFA optimization problem, we get the following formulation.

$$f(\Lambda, \Psi) = \arg \min_{\Lambda, \Psi} \|\text{Cov}(\mathbf{Y}) - \hat{\Sigma}\|$$

$$f(\tau, \alpha) = \arg \min_{\tau, \alpha} \|\bar{\mathbf{Y}} - \hat{\mu}\|$$



# MODEL IDENTIFICATION

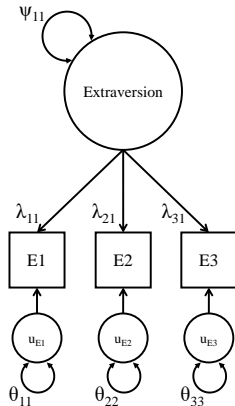


# Number of Estimated Parameters

To estimate a CFA model, we must define  $\hat{\Sigma}$ .

- We must estimate each parameter that defines an element in  $\hat{\Sigma}$ .
- The constraints on our model determine how many parameter estimates we need.

$$\hat{\Sigma} = \begin{bmatrix} \lambda_{11}\psi_{11}\lambda_{11} + \theta_{11} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \\ \lambda_{11}\psi_{11}\lambda_{21} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \\ \lambda_{11}\psi_{11}\lambda_{31} & \lambda_{21}\psi_{11}\lambda_{31} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \end{bmatrix}$$

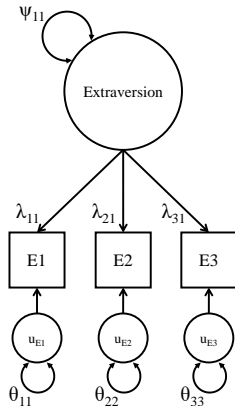


# Number of Estimated Parameters

For this example, we need to estimate seven parameters to fully define  $\hat{\Sigma}$ .

- One latent variance:  $\psi_{11}$
- Three factor loadings:  $\{\lambda_{11}, \lambda_{21}, \lambda_{31}\}$
- Three residual variances:  $\{\theta_{11}, \theta_{22}, \theta_{33}\}$

$$\hat{\Sigma} = \begin{bmatrix} \lambda_{11}\psi_{11}\lambda_{11} + \theta_{11} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \\ \lambda_{11}\psi_{11}\lambda_{21} & \lambda_{21}\psi_{11}\lambda_{21} + \theta_{22} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \\ \lambda_{11}\psi_{11}\lambda_{31} & \lambda_{21}\psi_{11}\lambda_{31} & \lambda_{31}\psi_{11}\lambda_{31} + \theta_{33} \end{bmatrix}$$



# Available Information

---

The data only contain a fixed amount of information.

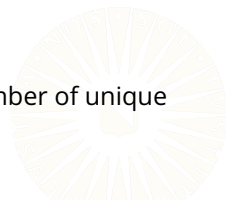
- We can quantify the available information in discrete units.
- Every unique element of  $\text{Cov}(\mathbf{Y})$  contributes one unit of information.

$$\text{Cov}(\mathbf{Y}) = \begin{bmatrix} \text{var}(y_1) & & \\ \text{cov}(y_2, y_1) & \text{var}(y_2) & \\ \text{cov}(y_3, y_1) & \text{cov}(y_3, y_2) & \text{var}(y_3) \end{bmatrix}$$

In this example, we have six pieces of available information.

- Three variances:  $\text{var}(y_1)$ ,  $\text{var}(y_2)$ ,  $\text{var}(y_3)$
- Three covariances:  $\text{cov}(y_2, y_1)$ ,  $\text{cov}(y_3, y_1)$ ,  $\text{cov}(y_3, y_2)$

For a positive-definite  $M \times M$  covariance matrix, the number of unique elements will always be  $Q = \frac{M(M+1)}{2}$ .





# Degrees of Freedom

---

We can only estimate one parameter for each piece of available information,  $Q$ .

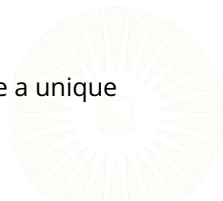
- We can estimate no more than  $Q$  parameters in any one model.

The *degrees of freedom* ( $df$ ) is the difference between the amount of information available in the data,  $Q$ , and the number of parameters estimated in our model,  $P$ .

$$df = Q - P$$

If  $df < 0$ , the model is not estimable.

- A model with  $df < 0$  is *not identified*.
- The data do not provide enough information to define a unique solution for all  $P$  parameter estimates.



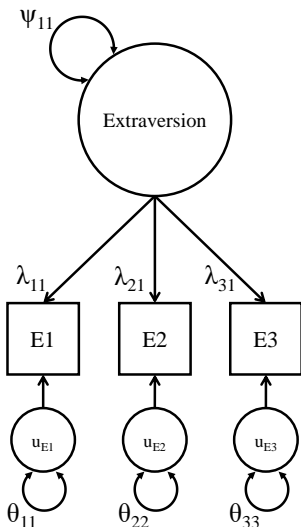
# Degrees of Freedom

What are the degrees of freedom for our example?

$$df = Q - P = 6 - 7 = -1$$

This model has negative  $df$ .

- We cannot estimate the model in this form.
- We must impose *identifying constraints*.



# Identifying Constraints

---

Consider the following equation:

$$5 = x + y$$

What are the values of  $x$  and  $y$ ?



# Identifying Constraints

---

Consider the following equation:

$$5 = x + y$$

What are the values of  $x$  and  $y$ ?

$$y = 5 - x$$



# Identifying Constraints

---

Consider the following equation:

$$5 = x + y$$

What are the values of  $x$  and  $y$ ?

$$y = 5 - x$$

What if we assume that  $y = x$ ?



# Identifying Constraints

---

Consider the following equation:

$$5 = x + y$$

What are the values of  $x$  and  $y$ ?

$$y = 5 - x$$

What if we assume that  $y = x$ ?

$$5 = x + y$$

$$0 = x - y$$



# Identifying Constraints

---

Consider the following equation:

$$5 = x + y$$

What are the values of  $x$  and  $y$ ?

$$y = 5 - x$$

What if we assume that  $y = x$ ?

$$5 = x + y$$

$$0 = x - y$$

Now we have enough information:

$$5 = x + x = 2x \Rightarrow x = y = 2.5$$



# Identifying Constraints

---

We must fix some parameters to identify the model.

- For each construct, we need  $df \geq 0$ .
- If the construct has three or more indicators:
  - Fix one parameter in the covariance model.
  - Fix one parameter in the mean model.
- If the construct has two indicators:
  - Fix an additional parameter in the covariance model.
- If the construct has only one indicator:
  - Cannot define a latent factor.





# Identifying Constraints

---

These constraints also define the scale of the latent factors.

- Latent factors have no direct representation as observed variables in our dataset.
- A latent factor only exists after we've estimated it.
- So latent factors have no inherent scale.
- Identifying constraints are also called *scaling constraints*.

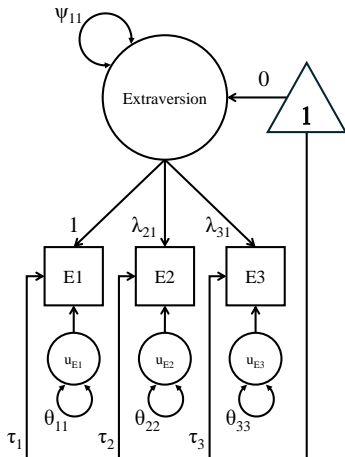
There are two common methods of identifying/scaling CFA models.

1. Marker-variable method
2. Fixed-factor method



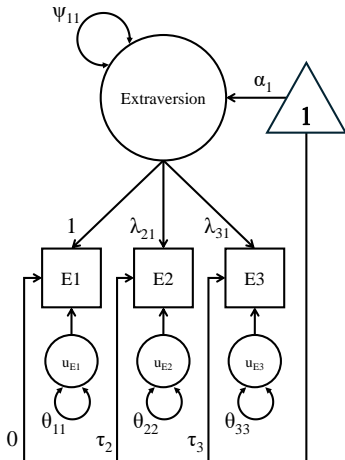
# Marker-Variable Method

- Fix one factor loading to 1.
- Fix the latent mean to 0.



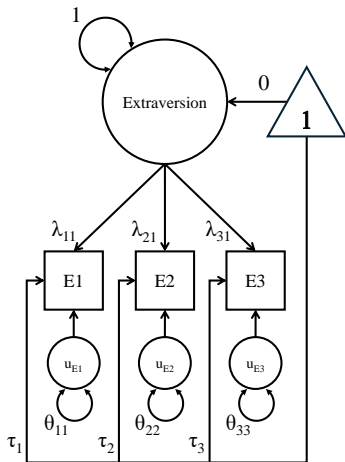
# Marker-Variable Method

- Fix one factor loading to 1.
- Estimate the latent mean and fix one item intercept to 0.



# Fixed-Factor Method

- Fix the latent variance to 1.
- Fix the latent mean to 0.



# Just-Identified: Unconstrained Model

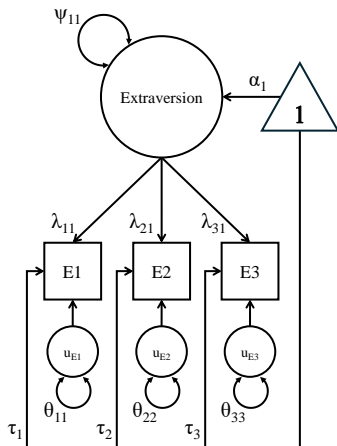
$$\alpha = [\alpha_1] \quad \Psi = [\psi_{11}]$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & \\ 0 & \theta_{22} & \\ 0 & 0 & \theta_{33} \end{bmatrix}$$

$$Q = \frac{3(3+1)}{2} + 3 = 9$$

$$\begin{aligned} df &= Q - P \\ &= 9 - 11 = -2 \end{aligned}$$



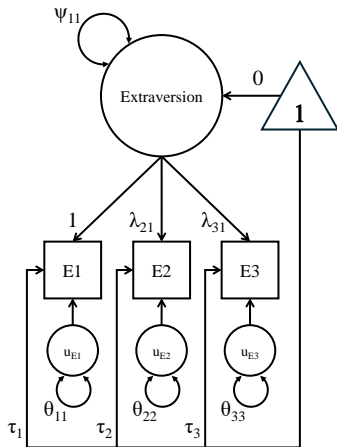
# Just-Identified: Marker-Variable

$$\alpha = [0] \quad \Psi = [\psi_{11}]$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 1 \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & \\ 0 & \theta_{22} & \\ 0 & 0 & \theta_{33} \end{bmatrix}$$

$$\begin{aligned} df &= Q - P \\ &= 9 - 9 = 0 \end{aligned}$$



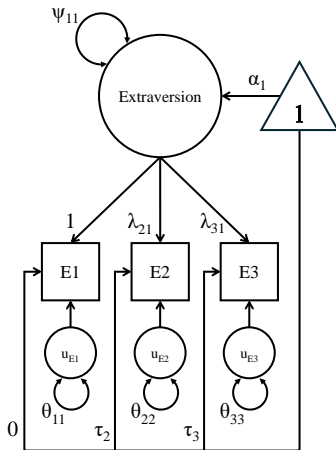
# Just-Identified: Marker-Variable

$$\alpha = [\alpha_1] \quad \Psi = [\psi_{11}]$$

$$\tau = \begin{bmatrix} 0 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 1 \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & \\ 0 & \theta_{22} & \\ 0 & 0 & \theta_{33} \end{bmatrix}$$

$$\begin{aligned} df &= Q - P \\ &= 9 - 9 = 0 \end{aligned}$$



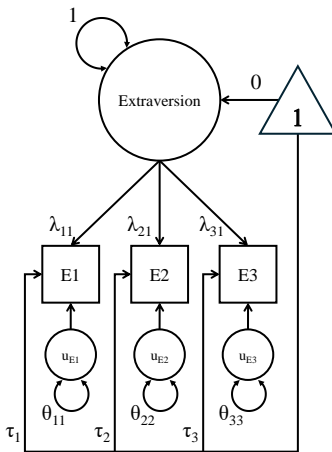
# Just-Identified: Fixed-Factor

$$\alpha = [0] \quad \Psi = [\psi_{11}]$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \\ \lambda_{31} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & & \\ 0 & \theta_{22} & \\ 0 & 0 & \theta_{33} \end{bmatrix}$$

$$\begin{aligned} df &= Q - P \\ &= 9 - 9 = 0 \end{aligned}$$





# Under-Identified: Unconstrained Model

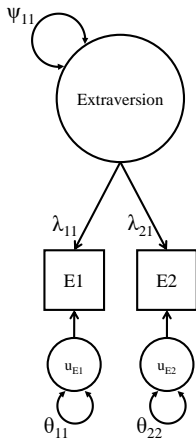
$$\Psi = [\psi_{11}]$$

$$\Lambda = \begin{bmatrix} \lambda_{11} \\ \lambda_{21} \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & \\ 0 & \theta_{22} \end{bmatrix}$$

$$Q = \frac{2(2+1)}{2} = 3$$

$$\begin{aligned} df &= Q - P \\ &= 3 - 5 = -2 \end{aligned}$$



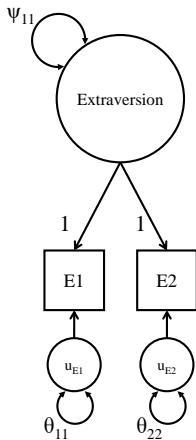
# Under-Identified: Marker-Variable

$$\Psi = [\psi_{11}]$$

$$\Lambda = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & \\ 0 & \theta_{22} \end{bmatrix}$$

$$\begin{aligned} df &= Q - P \\ &= 3 - 3 = 0 \end{aligned}$$



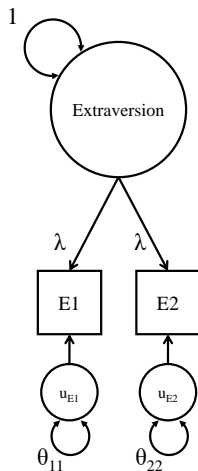
# Under-Identified: Fixed-Factor

$$\Psi = [1]$$

$$\Lambda = \begin{bmatrix} \lambda \\ \lambda \end{bmatrix}$$

$$\Theta = \begin{bmatrix} \theta_{11} & \\ 0 & \theta_{22} \end{bmatrix}$$

$$\begin{aligned} df &= Q - P \\ &= 3 - 3 = 0 \end{aligned}$$



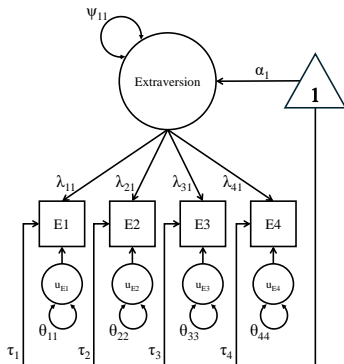
# Over-Identified Models

What happens when we have four (or more) indicators?

With 4 indicators, our model contains 14 parameters:

- Four factor loadings
- Four residual variances
- Four item intercepts
- One latent mean
- One latent variance

$$Q = \frac{4(4+1)}{2} + 4 = 14$$
$$df = 14 - 14 = 0$$



# Over-Identified Models

---

With 5 indicators, we have 17 model parameters:

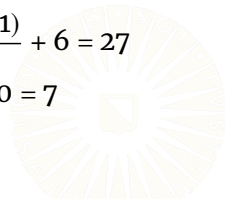
- Five factor loadings
- Five residual variances
- Five item intercepts
- One latent mean
- One latent variance

$$Q = \frac{5(5+1)}{2} + 5 = 20$$
$$df = 20 - 17 = 3$$

With 6 indicators, we have 20 model parameters:

- Six factor loadings
- Six residual variances
- Six item intercepts
- One latent mean
- One latent variance

$$Q = \frac{6(6+1)}{2} + 6 = 27$$
$$df = 27 - 20 = 7$$



# Over-Identified Models

With four indicators, we automatically have  $df \geq 0$  without imposing any scaling constraints.

- Can we directly estimate the unconstrained model?

```
mod1 <- '  
fun =~ Q77 + Q84 + Q170 + Q196  
'
```

```
fit1 <- lavaan(mod1,  
               data      = sapi,  
               auto.var  = TRUE,  
               meanstructure = TRUE,  
               int.ov.free = TRUE,  
               int.lv.free = TRUE)
```

```
Warning: lavaan->lav_model_vcov():  
  Could not compute standard errors! The information matrix could not be  
  inverted. This may be a symptom that the model is not identified.
```

Hmmm...I guess not.

# Over-Identified Models

---

```
partSummary(fit1, 1:4)
```

```
lavaan 0.6-19 ended normally after 13 iterations
```

Estimator	ML	
Optimization method	NLMINB	
Number of model parameters	14	
	Used	Total
Number of observations	970	1000

```
Model Test User Model:
```

Test statistic	58.017
Degrees of freedom	0

# Over-Identified Models

```
partSummary(fit1, 7:8)
```

Latent Variables:

	Estimate	Std.Err	z-value	P(> z )
fun =~				
Q77	0.824	NA		
Q84	0.578	NA		
Q170	0.479	NA		
Q196	0.619	NA		

Intercepts:

	Estimate	Std.Err	z-value	P(> z )
.Q77	3.574	NA		
.Q84	3.232	NA		
.Q170	3.955	NA		
.Q196	3.803	NA		
fun	0.000	NA		



# Over-Identified Models

---

```
partSummary(fit1, 9)
```

Variances:

	Estimate	Std.Err	z-value	P(> z )
.Q77	0.491	NA		
.Q84	0.760	NA		
.Q170	0.703	NA		
.Q196	0.370	NA		
fun	1.012	NA		



# Over-Identified Models

In the above example, the data contain enough information to define our model parameters, but that's not enough.

- We still need scaling constraints.
- The estimation algorithm needs an anchor point from which to extrapolate the relative values of the model parameters.
- Without any scaling constraints, an infinite number of parameter matrices will produce the same  $\hat{\mu}$  and  $\hat{\Sigma}$ .
- An infinite number of solutions are equally good.

```
## Fit a model to get some example parameters:
mod1 <- '
fun    =~ Q77 + Q84 + Q170 + Q196
liked  =~ Q44 + Q63 + Q76  + Q98
'

fit1 <- cfa(mod1, data = sapi, effect.coding = TRUE)
```

# Over-Identified Models

---

*## Extract the estimated factor loadings and latent covariance matrix:*

```
(lambda <- inspect(fit1, "est")$lambda)
```

```
      fun liked
Q77  1.254 0.000
Q84  0.954 0.000
Q170 0.795 0.000
Q196 0.997 0.000
Q44   0.000 0.764
Q63   0.000 1.155
Q76   0.000 1.133
Q98   0.000 0.949
```

```
(psi <- inspect(fit1, "est")$psi)
```

```
      fun liked
fun    0.399
liked 0.241 0.311
```

# Over-Identified Models

---

```
## Extract the estimated residual variances:
```

```
(theta <- inspect(fit1, "est")$theta)
```

	Q77	Q84	Q170	Q196	Q44	Q63	Q76	Q98
Q77	0.548							
Q84	0.000	0.727						
Q170	0.000	0.000	0.687					
Q196	0.000	0.000	0.000	0.364				
Q44	0.000	0.000	0.000	0.000	0.662			
Q63	0.000	0.000	0.000	0.000	0.000	0.807		
Q76	0.000	0.000	0.000	0.000	0.000	0.000	0.966	
Q98	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.469



# Over-Identified Models

---

```
## Manually compute the model-implied covariance matrix:
```

```
(sigma <- lambda %*% psi %*% t(lambda) + theta)
```

	Q77	Q84	Q170	Q196	Q44	Q63	Q76	Q98
Q77	1.176							
Q84	0.477	1.090						
Q170	0.398	0.303	0.940					
Q196	0.499	0.380	0.316	0.761				
Q44	0.231	0.175	0.146	0.183	0.843			
Q63	0.349	0.265	0.221	0.277	0.275	1.223		
Q76	0.342	0.260	0.217	0.272	0.269	0.407	1.365	
Q98	0.287	0.218	0.182	0.228	0.226	0.341	0.335	0.750



# Over-Identified Models

---

```
## Randomly sample an arbitrary scaling factor:
(a <- runif(1, 1, 2))

[1] 1.524314

## Rescale all factor loadings by a factor of a:
(lambda2 <- lambda * a)

      fun liked
Q77  1.911 0.000
Q84  1.454 0.000
Q170 1.212 0.000
Q196 1.520 0.000
Q44   0.000 1.164
Q63   0.000 1.760
Q76   0.000 1.727
Q98   0.000 1.447
```

# Over-Identified Models

```
## Rescale the latent covariance matrix by a factor of (1 / a^2):
(psi2 <- psi / a^2)

      fun liked
fun    0.172
liked 0.104 0.134

## Compute the model-implied covariance matrix using the rescaled parameters:
(sigma2 <- lambda2 %*% psi2 %*% t(lambda2) + theta)

      Q77    Q84   Q170   Q196    Q44    Q63    Q76    Q98
Q77    1.176
Q84    0.477 1.090
Q170   0.398 0.303 0.940
Q196   0.499 0.380 0.316 0.761
Q44    0.231 0.175 0.146 0.183 0.843
Q63    0.349 0.265 0.221 0.277 0.275 1.223
Q76    0.342 0.260 0.217 0.272 0.269 0.407 1.365
Q98    0.287 0.218 0.182 0.228 0.226 0.341 0.335 0.750
```

# Over-Identified Models

```
## Compare the two model-implied covariance matrices:
```

```
all.equal(sigma, sigma2)
```

```
[1] TRUE
```

```
## Repeat the experiment 100 times:
```

```
out <- rep(NA, 100)
```

```
for(i in 1:100) {
```

```
  a <- runif(1, 1, 2)
```

```
  lambda2 <- lambda * a
```

```
  psi2 <- psi / a^2
```

```
  sigma2 <- lambda2 %*% psi2 %*% t(lambda2) + theta
```

```
  out[i] <- all.equal(sigma, sigma2)
```

```
}
```

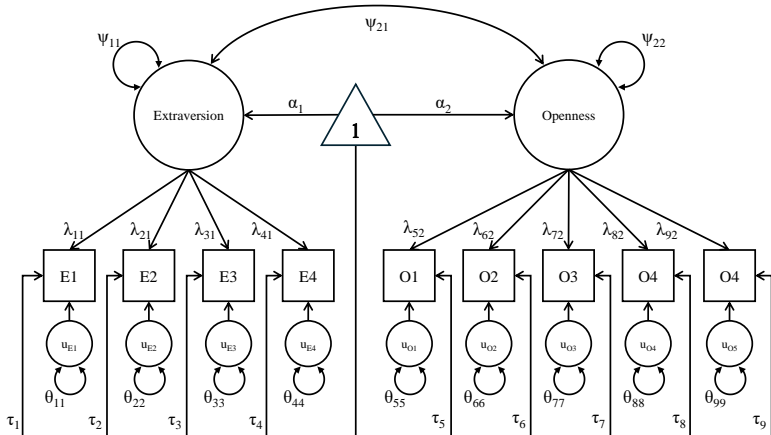
```
## Always the same result?
```

```
all(out)
```

```
[1] TRUE
```



# Two Construct: Unconstrained Model

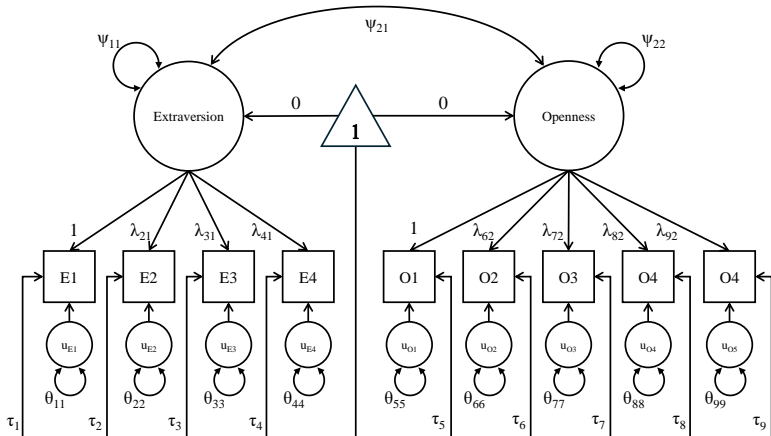


# Unconstrained Parameter Matrices

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \Psi = \begin{bmatrix} \psi_{11} & \\ \psi_{21} & \psi_{22} \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \\ \tau_8 \\ \tau_9 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$

# Two Construct: Marker variable

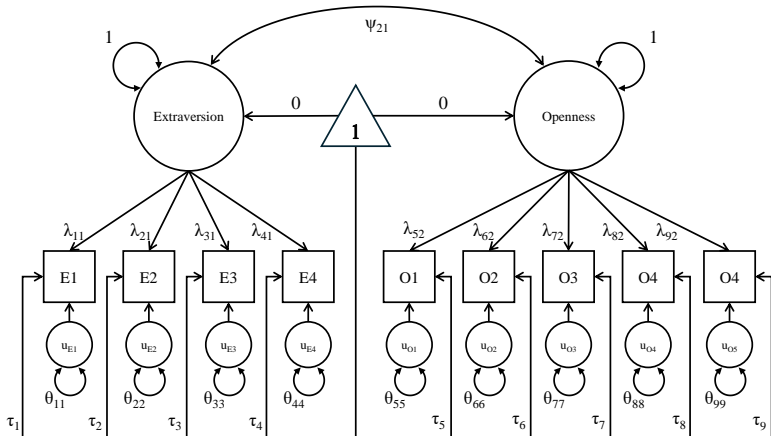


# Parameter Matrices: Marker Variable

$$\alpha = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Psi = \begin{bmatrix} \psi_{11} & \\ \psi_{21} & \psi_{22} \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \\ \tau_8 \\ \tau_9 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix} \quad \Lambda = \begin{bmatrix} 1 & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & 1 \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$

# Two Construct: Fixed-Factor



# Parameter Matrices: Fixed-Factor

$$\alpha = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \Psi = \begin{bmatrix} 1 & \\ \psi_{21} & 1 \end{bmatrix}$$

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \\ \tau_7 \\ \tau_8 \\ \tau_9 \end{bmatrix} \quad \Theta = \begin{bmatrix} \theta_{11} & & & & & & & & \\ 0 & \theta_{22} & & & & & & & \\ 0 & 0 & \theta_{33} & & & & & & \\ 0 & 0 & 0 & \theta_{44} & & & & & \\ 0 & 0 & 0 & 0 & \theta_{55} & & & & \\ 0 & 0 & 0 & 0 & 0 & \theta_{66} & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_{77} & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{88} & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \theta_{99} \end{bmatrix} \quad \Lambda = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \\ 0 & \lambda_{92} \end{bmatrix}$$

# EXAMPLE

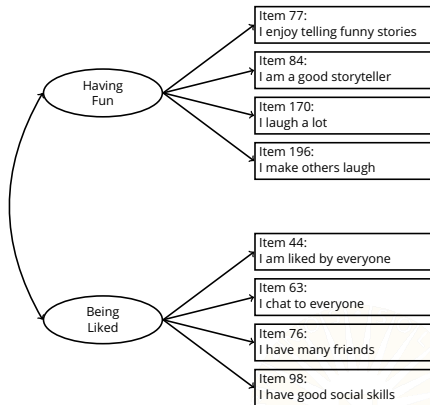


# CFA of Extraversion Items

Suppose we hypothesize two distinct dimensions of extraversion underlying 8 of the SAPI items.

1. Having Fun
2. Being Liked by Others

We'll define our measurement model as the two-factor CFA shown to the right.





# Example: Marker Variable

Load the SAPI data.

```
dataDir <- "../data/"
sapi <- read.table(paste0(dataDir, "sapi.txt"),
                  header = TRUE,
                  na.strings = "-999")
```

Specify the **lavaan** model syntax for the SAPI extraversion CFA.

```
mod1 <- '
fun    =~ Q77 + Q84 + Q170 + Q196
liked =~ Q44 + Q63 + Q76  + Q98
'
```

Use the `cfa()` function to estimate the model.

```
library(lavaan)
out1 <- cfa(mod1, data = sapi)
```

# Example: Marker Variable

```
partSummary(out1, 1:4)
```

lavaan 0.6-19 ended normally after 30 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	17

	Used	Total
Number of observations	959	1000

Model Test User Model:

Test statistic	130.193
Degrees of freedom	19
P-value (Chi-square)	0.000

# Example: Marker Variable

```
partSummary(out1, 5:7)
```

Parameter Estimates:

Standard errors	Standard
Information	Expected
Information saturated (h1) model	Structured

Latent Variables:

	Estimate	Std.Err	z-value	P(> z )
fun =~				
Q77	1.000			
Q84	0.761	0.051	14.902	0.000
Q170	0.634	0.047	13.558	0.000
Q196	0.795	0.046	17.381	0.000
liked =~				
Q44	1.000			
Q63	1.512	0.147	10.278	0.000
Q76	1.483	0.149	9.955	0.000
Q98	1.243	0.119	10.462	0.000

# Example: Marker Variable

```
partSummary(out1, 8:9)
```

Covariances:

	Estimate	Std.Err	z-value	P(> z )
fun ~~				
liked	0.231	0.025	9.234	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z )
.Q77	0.548	0.038	14.389	0.000
.Q84	0.727	0.039	18.703	0.000
.Q170	0.687	0.035	19.572	0.000
.Q196	0.364	0.025	14.731	0.000
.Q44	0.662	0.034	19.291	0.000
.Q63	0.807	0.048	16.943	0.000
.Q76	0.966	0.054	17.931	0.000
.Q98	0.469	0.029	16.121	0.000
fun	0.627	0.056	11.303	0.000
liked	0.182	0.029	6.290	0.000

# Example: Marker Variable

```
inspect(out1, "r2")
```

```
  Q77  Q84  Q170  Q196  Q44  Q63  Q76  Q98  
0.534 0.333 0.268 0.521 0.215 0.340 0.293 0.374
```

```
fitMeasures(out1) |> head(22) |> round(3)
```

npar	fmin	chisq	df
17.000	0.068	130.193	19.000
pvalue	baseline.chisq	baseline.df	baseline.pvalue
0.000	1574.886	28.000	0.000
cfi	tli	nnfi	rfi
0.928	0.894	0.894	0.878
nfi	pnfi	ifi	rni
0.917	0.622	0.929	0.928
logl	unrestricted.logl	aic	bic
-10147.587	-10082.491	20329.175	20411.895
ntotal	bic2		
959.000	20357.903		

# Example: Marker Variable

```
fitMeasures(out1) |> tail(-22) |> round(3)
```

rmsea	rmsea.ci.lower	rmsea.ci.upper
0.078	0.066	0.091
rmsea.ci.level	rmsea.pvalue	rmsea.close.h0
0.900	0.000	0.050
rmsea.notclose.pvalue	rmsea.notclose.h0	rmr
0.421	0.080	0.042
rmr_nomean	srmr	srmr_bentler
0.042	0.043	0.043
srmr_bentler_nomean	crmr	crmr_nomean
0.043	0.049	0.049
srmr_mplus	srmr_mplus_nomean	cn_05
0.043	0.043	223.037
cn_01	gfi	agfi
267.582	0.968	0.939
pgfi	mfi	ecvi
0.511	0.944	0.171

# Example: Fixed Factor

---

We only need to change one option to implement the fixed-factor method.

- The `std.lv = TRUE` option (i.e., standardized latent variables) applies the appropriate constraints.

```
out2 <- cfa(mod1, data = sapi, std.lv = TRUE)
```



# Example: Fixed Factor

```
partSummary(out2, 1:4)
```

lavaan 0.6-19 ended normally after 17 iterations

Estimator	ML	
Optimization method	NLMINB	
Number of model parameters	17	
	Used	Total
Number of observations	959	1000

Model Test User Model:

Test statistic	130.193
Degrees of freedom	19
P-value (Chi-square)	0.000



# Example: Fixed Factor

```
partSummary(out2, 5:7)
```

Parameter Estimates:

Standard errors	Standard
Information	Expected
Information saturated (h1) model	Structured

Latent Variables:

	Estimate	Std.Err	z-value	P(> z )
fun =~				
Q77	0.792	0.035	22.606	0.000
Q84	0.603	0.035	17.193	0.000
Q170	0.502	0.033	15.180	0.000
Q196	0.630	0.028	22.308	0.000
liked =~				
Q44	0.426	0.034	12.580	0.000
Q63	0.644	0.040	16.071	0.000
Q76	0.632	0.043	14.845	0.000
Q98	0.530	0.031	16.912	0.000

# Example: Fixed Factor

```
partSummary(out2, 8:9)
```

Covariances:

	Estimate	Std.Err	z-value	P(> z )
fun ~~				
liked	0.683	0.033	20.483	0.000

Variances:

	Estimate	Std.Err	z-value	P(> z )
.Q77	0.548	0.038	14.389	0.000
.Q84	0.727	0.039	18.703	0.000
.Q170	0.687	0.035	19.572	0.000
.Q196	0.364	0.025	14.731	0.000
.Q44	0.662	0.034	19.291	0.000
.Q63	0.807	0.048	16.943	0.000
.Q76	0.966	0.054	17.931	0.000
.Q98	0.469	0.029	16.121	0.000
fun	1.000			
liked	1.000			

# Example: Fixed Factor

```
inspect(out2, "r2")
```

```
      Q77   Q84  Q170  Q196   Q44   Q63   Q76   Q98  
0.534 0.333 0.268 0.521 0.215 0.340 0.293 0.374
```

```
fitMeasures(out2) |> head(22) |> round(3)
```

npar	fmin	chisq	df
17.000	0.068	130.193	19.000
pvalue	baseline.chisq	baseline.df	baseline.pvalue
0.000	1574.886	28.000	0.000
cfi	tli	nnfi	rfi
0.928	0.894	0.894	0.878
nfi	pnfi	ifi	rni
0.917	0.622	0.929	0.928
logl	unrestricted.logl	aic	bic
-10147.587	-10082.491	20329.175	20411.895
ntotal	bic2		
959.000	20357.903		

# Example: Fixed Factor

```
fitMeasures(out2) |> tail(-22) |> round(3)
```

rmsea	rmsea.ci.lower	rmsea.ci.upper
0.078	0.066	0.091
rmsea.ci.level	rmsea.pvalue	rmsea.close.h0
0.900	0.000	0.050
rmsea.notclose.pvalue	rmsea.notclose.h0	rmr
0.421	0.080	0.042
rmr_nomean	srmr	srmr_bentler
0.042	0.043	0.043
srmr_bentler_nomean	crmr	crmr_nomean
0.043	0.049	0.049
srmr_mplus	srmr_mplus_nomean	cn_05
0.043	0.043	223.037
cn_01	gfi	agfi
267.582	0.968	0.939
pgfi	mfi	ecvi
0.511	0.944	0.171

# Compare Results

```
inspect(out1, "est")$lambda - inspect(out2, "est")$lambda
```

```
      fun liked  
Q77  0.208 0.000  
Q84  0.158 0.000  
Q170 0.132 0.000  
Q196 0.165 0.000  
Q44  0.000 0.574  
Q63  0.000 0.868  
Q76  0.000 0.851  
Q98  0.000 0.713
```

```
inspect(out1, "est")$psi - inspect(out2, "est")$psi
```

```
      fun liked  
fun    -0.373  
liked -0.453 -0.818
```

# Compare Results

```
all.equal(fitMeasures(out1), fitMeasures(out2))
```

```
[1] TRUE
```

```
inspect(out1, "r2") - inspect(out2, "r2")
```

Q77	Q84	Q170	Q196	Q44	Q63	Q76	Q98
0	0	0	0	0	0	0	0

```
inspect(out1, "est")$theta - inspect(out2, "est")$theta
```

	Q77	Q84	Q170	Q196	Q44	Q63	Q76	Q98
Q77	0							
Q84	0	0						
Q170	0	0	0					
Q196	0	0	0	0				
Q44	0	0	0	0	0			
Q63	0	0	0	0	0	0		
Q76	0	0	0	0	0	0	0	
Q98	0	0	0	0	0	0	0	0