Latent Trait Measurement Models for Binary Responses: Part 2

- Topics:
  - Review of IRT parameters (and choosing between models)
  - From Item Response Theory to Item Factor Analysis
  - Model estimation, comparison, and evaluation

IRT Models for Binary Responses

- Range from 1–4 item parameters that predict the link-transformed probability of $y_{is} = 1$ (correct or endorsed)
  - Link = logit → natural log of the odds of the probability (of a 1)
  - Link = probit = ogive → z-score for area to the left of the probability under a standard normal distribution (inverse link requires integration)
  - No estimated residual variances → Var(y_i^*) = 3.29 for logit, 1.00 for probit
  - Latent factor (per subject) is now called theta (θ), but it's the same idea

- Estimated parameters (as fixed effects) per item:
  - $\boldsymbol{b_i}$ = difficulty → location on theta latent trait
  - $\boldsymbol{a_i}$ = slope → discrimination → relation to trait at b location
    - In "Rasch" models, items differ only in difficulty (and common a slope = 1)

IRT → IFA above the line below only

  - $\boldsymbol{c_i}$ = lower asymptote → guessing → lowest possible probability
  - $\boldsymbol{d_i}$ = upper asymptote → carelessness → highest possible probability
    - In multidimensional IRT models, d is used for an intercept instead (I know, I'm sorry)

Model Comparisons in IRT: Decide Between Models via −2ΔLL Tests

- **Nested models** can be compared with the same −2ΔLL tests we used in CFA → without the "robust" part of ML, so they get simpler (scale factor=1)
  - e.g., Does a 2-parameter model fit better than a 1-parameter model?
  - Step 1: Calculate $-2\Delta LL = -2(LL_{fewer} - LL_{more})$
  - Step 2: Calculate $\Delta df = df_{more} - df_{fewer}$ (given as "# free parms")
  - Compare −2ΔLL with df = Δdf to χ^2 critical values (or excel CHIDIST)

- If **adding** parameters, model fit can get **better** or **not better**
- If **removing** parameters, model fit can get **worse** or **not worse**
- **AIC and BIC** values (from −2LL) can be used to compare non-nested models (given same sample of subjects and items), **smaller is better**
- Models with different items are still not comparable by −2LL, AIC, or BIC!!!
- Assessing global and local model fit can be much trickier… stay tuned!

Item Response Theory (IRT) = Item Factor Analysis (IFA) Models

Mplus can do ALL of these model/estimator combinations:	Model form: with **discrimination** and **difficulty** parameters	Model form: with **loadings** and **threshold** parms
Full-information estimation via Maximum Likelihood ("Marginal ML") → *uses original item responses*	**"IRT"** (Mplus gives only for binary responses)	**"?"** (Mplus gives for all models)
Limited-information estimation via Weighted Least Squares ("WLSMV") → *uses item response summary*	**"?"** (Mplus gives only for binary responses)	**"IFA"** (Mplus gives for all models)

- CFA assumes normally distributed, continuous item responses, but "**CFA** models for **categorical** responses" = **IRT** and **IFA** models

- These different names are used to reflect the combination of how the model is specified and how it is estimated, but it's the same core model
  - Btw, R Lavaan only has limited-information estimation for these models… (so use MIRT)

Relating Item Factor Analysis (IFA) to 2-P Item Response Models (IRT)

- CFA → linear regression as IRT → logistic regression **i = item, s = subject**
  - Predictor x_s is observed, but predictor F_s is latent (*aka*, factor, variable, trait)

- **Linear regression model** → **CFA model** (for continuous responses):

$$y_{is} = \beta_{0i} + \beta_{1i} x_s + e_{is} \qquad y_{is} = \mu_i + \lambda_i F_s + e_{is}$$

- **Logistic regression model** (for 0/1 responses, so there is no e_{is} residual):

$$\text{Log}\left[\frac{p(y_{is}=1)}{p(y_{is}=0)}\right] = \beta_{0i} + \beta_{1i} x_s$$

Why does the IRT model below look so different than the CFA model? Here's how these models all relate…

- **2-PL IRT model** (for 0/1 responses, so there is no e_{is} residual):

$$\text{Log}\left[\frac{p(y_{is}=1)}{p(y_{is}=0)}\right] = a_i(\theta_s - b_i)$$

Relating Regression, CFA, IFA, and IRT

- **Linear regression model** and **(Linear) Confirmatory FA model**:

$$y_{is} = \beta_{0i} + \beta_{1i}x_s + e_{is} \qquad y_{is} = \mu_i + \lambda_i F_s + e_{is}$$

- **Binary regression models** and **Binary Item Factor Analysis models!**

$$\text{Logit}\,[p(y_{is}=1)] = \beta_{0i} + \beta_{1i}x_s \qquad \text{Logit}[p(y_{is}=1)] = -\tau_i + \lambda_i F_s$$
$$\text{Probit}[p(y_{is}=1)] = \beta_{0i} + \beta_{1i}x_s \qquad \text{Probit}[p(y_{is}=1)] = -\tau_i + \lambda_i F_s$$

- **Binary Item Response Theory models:**

2PL: $\text{Logit}\,[p(y_{is}=1)] = a_i(\theta_s - b_i)$

Ogive: $\text{Probit}[p(y_{is}=1)] = a_i(\theta_s - b_i)$

Logit to Probability:

$$\text{prob} = \frac{\exp(\text{logit})}{1+\exp(\text{logit})}$$

- In CFA, item loading $\boldsymbol{\lambda_i}$ → **discrimination** and item intercept $\boldsymbol{\mu_i}$ → **difficulty**, but difficulty was backwards (easier or less severe items had higher means)…
- In IFA for binary items within Mplus, the **intercept** $\boldsymbol{\mu_i}$ (which was really **easiness**) becomes a "**threshold**" $\boldsymbol{\tau_i}$ that really does index **difficulty**: $\boldsymbol{\mu_i = -\tau_i}$
  → this provides continuity of direction with the IRT b_i "difficulty" values
- The **2-P IRT and IFA models get re-arranged into each other** as follows…

From IFA to IRT

IFA with "easiness" **intercept** μ_i: $\mathbf{Logit\ or\ Probit}\ y_{is} = \mu_i + \lambda_i F_s \quad \mu_i = -\tau_i$

IFA with "difficulty" **threshold** τ_i: $\mathbf{Logit\ or\ Probit}\ y_{is} = -\tau_i + \lambda_i F_s$

IFA model with "difficulty" thresholds can be written as a **2-PL IRT Model:**

IRT model:

$\mathbf{Logit\ or\ Probit}\ y_{is} = a_i(\theta_s - b_i)$

IFA model:

$= \underbrace{-a_i b_i}_{\tau_i} + \underbrace{a_i \theta_s}_{\lambda_i}$

a_i = discrimination
b_i = difficulty
θ_s = F_s latent trait

Convert IFA to IRT:

$$a_i = \lambda_i * \sqrt{\text{theta variance}}$$

$$b_i = \frac{\tau_i - (\lambda_i * \text{theta mean})}{\lambda_i * \sqrt{\text{theta variance}}}$$

Convert IRT to IFA:

$$\lambda_i = \frac{a_i}{\sqrt{\text{theta variance}}}$$

$$\tau_i = a_i b_i + \frac{a_i * \text{theta mean}}{\sqrt{\text{theta variance}}}$$

Note: These formulas rescale a_i and b_i so that theta M=0, VAR=1

If you don't want to rescale theta, use M=0 and VAR=1 for it to keep your current scale

Thus, IFA = 2-P IRT, just re-arranged!

2-P IRT: **IFA:**

$$\textbf{Logit or Probit } y_{is} = a_i(\theta_s - b_i) = \underbrace{-a_i b_i}_{\tau_i} + \underbrace{a_i \theta_s}_{\lambda_i}$$

- An item factor model for binary outcomes is the same as a two-parameter IRT model, so you can keep both camps happy:
  - IFA loadings λ_i can be converted into 2-P IRT discriminations a_i
  - IFA thresholds $\tau_i = -\mu_i$ can be converted into 2-P IRT difficulties b_i

---

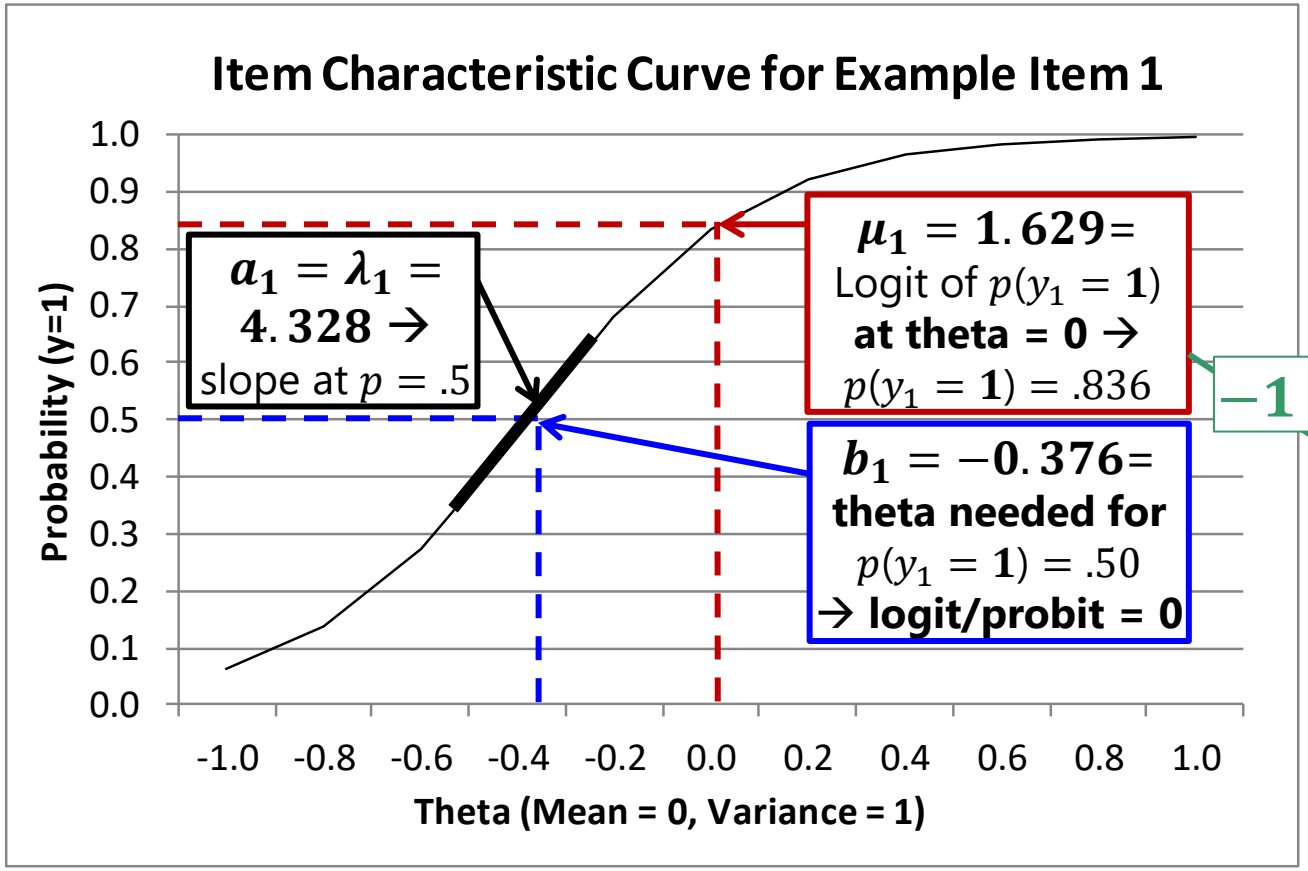
- CFA/SEM crowd? **Use logit or probit** $y_{is} = -\tau_i + \lambda_i F_s$
  - "**I used IFA**" → Report item "factor loadings" λ_i and "thresholds" τ_i
  - See also "**CFA for categorical data**" as usually synonymous
- IRT crowd? **Use logit or probit** $y_{is} = a_i(\theta_s - b_i)$
  - "**I used IRT**" → Report item "discriminations" a_i and "difficulties" b_i

Item Parameter Interpretations

IFA model with **loading** and "easiness" **intercept** μ_i: $\text{logit}\, y_{is} = \mu_i + \lambda_i F_s$

IFA model with **loading** and "difficulty" **threshold** τ_i: $\text{logit}\, y_{is} = -\tau_i + \lambda_i F_s$

2-P IRT model with **discrimination** and **difficulty** b_i: $\text{logit}\, y_{is} = a_i(\theta_s - b_i)$



From IRT to IFA if theta M=0, SD=1:

$\lambda_i = a_i$

$\tau_i = a_i b_i$

Item Parameter Interpretations

IFA model with **loading** and "easiness" **intercept** $\boldsymbol{\mu_i}$: $\textbf{logit}\, \boldsymbol{y_{is}} = \boldsymbol{\mu_i} + \boldsymbol{\lambda_i F_s}$

IFA model with **loading** and "difficulty" **threshold** $\boldsymbol{\tau_i}$: $\textbf{logit}\, \boldsymbol{y_{is}} = -\boldsymbol{\tau_i} + \boldsymbol{\lambda_i F_s}$

2-P IRT model with **discrimination** and **difficulty** $\boldsymbol{b_i}$: $\textbf{logit}\, \boldsymbol{y_{is}} = \boldsymbol{a_i}(\boldsymbol{\theta_s} - \boldsymbol{b_i})$

- IFA and 2-P IRT item slope parameters are interpreted similarly:
  - IFA loading λ_i = Δ in logit/probit of $y_{is} = 1$ per unit Δ in theta
  - IRT discrimination a_i = slope of ICC at prob=.50 (where logit/probit = 0)

- IFA and 2-P IRT item location parameters are different:
  - **IFA intercept** $\boldsymbol{\mu_i}$ = logit/probit of $\boldsymbol{y_{is} = 1}$ when **theta (x) = 0**
  - **IFA threshold** $\boldsymbol{\tau_i}$ = logit/probit of $\boldsymbol{y_{is} = 0}$ when **theta (x) = 0**
  - **IRT difficulty** $\boldsymbol{b_i}$ = amount of theta needed for **logit/probit (y) = 0**
    - So $\boldsymbol{b_i}$ difficulty values are more useful (to me) to index **location**

3 Kinds of 2-P Model Output in Mplus

- **IFA unstandardized solution:**
  - **Item threshold $\boldsymbol{\tau_i}$** = expected logit/probit of $\boldsymbol{y_{is} = 0}$ when theta $\boldsymbol{\theta_s = 0}$
  - **Item loading $\boldsymbol{\lambda_i}$** = Δ in logit/probit of $\boldsymbol{y_{is} = 1}$ per unit Δ in $\boldsymbol{\theta_s}$ (theta)
  - Item residual variance not estimated, but is 3.29 in logit or 1.00 in probit for y_{is}^*

- **IFA standardized solution:**
  - Total variance of logit or probit of $y_i = 1$ → (λ_i^2 * theta variance) + (3.29 or 1)
  - **std $\boldsymbol{\tau_i}$** = unstd $\boldsymbol{\lambda_i}$ / SD(logit or probit of $y_i = 1$) → not usually interpreted
  - **std $\boldsymbol{\lambda_i}$** = unstd $\boldsymbol{\lambda_i}$ * SD(theta) / SD(logit or probit of $y_i = 1$)
    → Correlation of logit or probit of item response with theta

IFA solution **should not** be used to compute Omega!

- **IRT solution** (only one type; only given directly in Mplus for binary items):
  - $\boldsymbol{b_i}$ = theta at which $\text{prob}(y_{is} = 1)$ = .50 (where logit or probit = 0)
  - $\boldsymbol{a_i}$ = Δ in logit or probit of $y_{is} = 1$ per unit Δ in $\boldsymbol{\theta_s}$ (theta)
    = slope of item characteristic curve at $\boldsymbol{b_i}$ item difficulty location

CFA vs. IRT/IFA vs. ???

- CFA assumes continuous, normally distributed item responses
  - Robust ML can be used to adjust fit statistics and parameter SEs for non-normality, but it's still a **linear model** for the factor predicting y_{is}
  - A linear model may not be plausible for ordinal or other bounded responses (i.e., the model-predicted responses may extend beyond the possible response options for plausible ranges of values of the latent factor)

- IRT/IFA is for binary (or ordinal or nominal) item responses
  - **Linear model between theta and logit or probit of y_{is} instead**
  - Because ordinal item responses are bounded and are not really numbers, IRT/IFA should probably be used for these kinds of responses
  - CFA may not be too far off given ≥ 5 normally distributed responses, but then you can't see how useful your answer choices are (stay tuned!)

- For non-normal but continuous (not categorical) responses, other latent trait measurement models are possible (stay tuned!)

Summary: Binary IRT/IFA Models

- IRT/IFA are a family of models that specify the relationship between the latent trait ("theta") and a link-transformation of probability of $y_{is} = 1$
  - **Linear** relationship between theta and **logit or probit** of $y_{is} = 1$
    → **nonlinear** relationship between theta and **probability** of $y_{is} = 1$

- The form of the trait–response relationship depends on:
  - At least the location on the latent trait (given by difficulty $\boldsymbol{b_i}$ or threshold $\boldsymbol{\tau_i}$)
  - Strength of relationship with theta; may vary across items (given by $\boldsymbol{a_i}$ or $\boldsymbol{\lambda_i}$)
    - If not, its a "1-P" or "Rasch model" → assumes tau-equivalence (equal discrimination)
  - Also maybe lower and upper asymptotes ($\boldsymbol{c_i}$ and $\boldsymbol{d_i}$) → but good luck with that!

- Because the loadings/slopes relate nonlinearly to theta, this implies that **reliability** (now called "test information") **must vary** across theta values
  - So items are not just "good" or "bad", but are "good" or "bad" for whom?

- **Now what about model fit??? We have to talk estimation first…**

What all do we have to estimate?

- For example, a 7-item binary test and a 2-PL model, (assuming we fix the theta distribution to have mean=0 and variance=1):
  - 7 item discriminations ($\boldsymbol{a_i}$) and 7 item difficulties ($\boldsymbol{b_i}$) = 14 parameters

- **Item parameters** are **FIXED effects** → specific item inference
  - Fixed effects do not have a distribution (at least in frequentist-land)

- What about the all the individual subject **thetas**?
  - These factor scores are <u>not</u> part of the model likelihood—thetas are **RANDOM effects** (= U's in multilevel, btw) that have a distribution
  - Thus, our inference is about the distribution of the latent traits in the population of subjects, which we assume to be multivariate normal
  - So we will need the **theta means, variances, and covariances** for the sample, but **not** the theta estimates for each **subject** per se

Estimation: Items, then Subjects

3 full-information item estimation methods:

- "**Full-information**" → uses individual item responses
- 3 methods differ with respect to how they handle unknown subject thetas
- First, two less-used and older methods:
  - **"Conditional" ML** → *theta? We don't need no stinkin' theta…*
    - Uses total score as "theta" (so can't include subject with all 0 or 1 responses)
    - Thus, is only possible within Rasch models (where the total is sufficient for theta)
    - If the Rasch model holds, estimators are consistent and efficient and can be treated like true likelihood values (i.e., can be used in model comparisons)
  - **"Joint" ML** → *Um, can we just pretend the thetas are fixed effects instead?*
    - Iterates back and forth between subjects and items (each as fixed effects) until item parameters don't change much—then calls it done (i.e., converged)
    - Many disadvantages: estimators are biased, inconsistent, with too small SEs and likelihoods that can't be used in model comparisons
    - More subjects → more parameters to estimate, too → so bad gets even worse!

Marginal ML Estimation (with Numeric Integration)

- Gold standard of estimation (used in Mplus, but not lavaan!)
  - This is the same idea of multivariate height, just using a different distribution than multivariate normal for the log-likelihood function
- Relies on two assumptions of **independence**:
  - Item responses are "locally" independent after controlling for theta
    - This means that the joint probability (likelihood) of two item responses is just the probability of each multiplied together
  - Subjects are independent (no clustering or nesting)
    - You can add random effects to capture dependency, but then the assumption is "independent after controlling for random effects"
- Doesn't assume it knows the individual thetas, but it does assume that the *distribution* of theta(s) is (multivariate) normal

Marginal ML via Numeric Integration

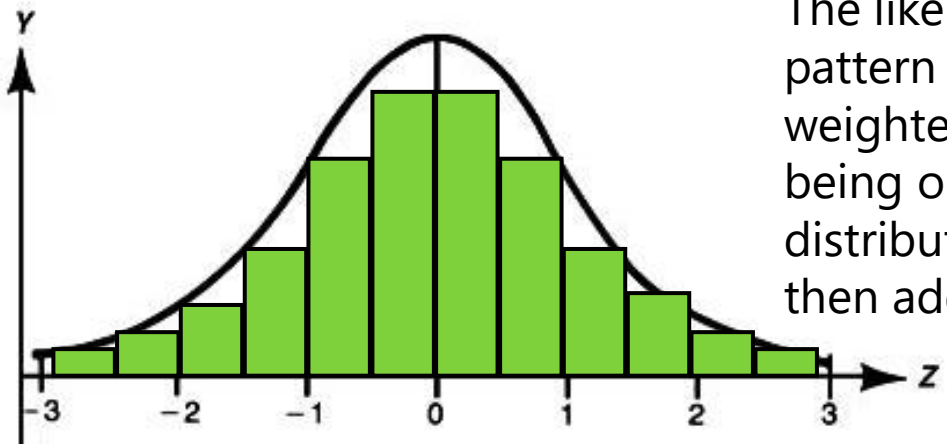
- **Step 1:** Select starting values for all item parameters (e.g., using CTT)
- **Step 2:** Compute the **likelihood for each subject** given by the *current* parameter values (using start values or updated values later on)
  - IRT model gives probability of response given item parameters and theta
  - To get likelihood per subject, take each predicted item probability and plug them into: **likelihood (all responses) = product over items of: $p^y(1-p)^{1-y}$**
  - But we don't have theta yet! No worries: computing the likelihood for each set of possible parameters requires *removing* the individual thetas from the model equation—by ***integrating*** across the possible theta values for each subject
  - Integration is done by "Gaussian Quadrature" → summing up rectangles that approximate the integral (the area under the curve) for each subject
- **Step 3:** Decide if you have the right answers, which occurs when the sum of the log-likelihoods changes very little across iterations (i.e., it converges)
- **Step 4:** If you aren't converged, choose new parameters values
  - Newton-Rhapson or Fisher Scoring (calculus), EM algorithm (thetas =missing data)

"Marginal" ML Estimation

- More on Step 2: Divide the theta distribution into rectangles

  → "**Gaussian Quadrature**" (# rectangles = # "**quadrature points**")

  - Divide the whole distribution into rectangles, and then take the most likely section for each subject and rectangle that more specifically
    - This is "**adaptive quadrature**" and is computationally more demanding, but gives more accurate results with fewer rectangles (Mplus uses 15)



The likelihood of each subject's response pattern at each theta rectangle is then weighted by that rectangle's probability of being observed (as given by the normal distribution). The weighted likelihoods are then added together across all rectangles.

→ ta da! "**numeric integration**"

  - Unfortunately, each additional theta or factor adds another dimension of integration (so 2 factors = 15*15 rectangles to try at each iteration)

Example of Numeric Integration

1. Start values for item parameters (here, $a = 1$ for convenience):
   - Item 1: mean = .73 → logit = +1, so starting $b_1 = -1$
   - Item 2: mean = .27 → logit = −1, so starting $b_2 = +1$
2. Compute per-subject likelihood using item parameters and set of thetas (e.g., −2,0,2) with IRT model: $\text{logit}(y_{is} = 1) = a(\theta - b_i)$

	Theta = -2	Logit	IF y=1 Prob	IF y=0 1-Prob	Likelihood if both y=1	Theta prob	Theta width	Product per Theta
Item 1 b = -1	(-2 - -1)	-1	0.27	0.73	0.0127548	0.05	2	0.001275
Item 2 b = +1	(-2 - 1)	-3	0.05	0.95				
	Theta = 0	**Logit**	**Prob**	**1-Prob**				
Item 1 b = -1	(0 - -1)	1	0.73	0.27	0.1966119	0.40	2	0.15729
Item 2 b = +1	(0 - 1)	-1	0.27	0.73				
	Theta = +2	**Logit**	**Prob**	**1-Prob**				
Item 1 b = -1	(2 - -1)	3	0.95	0.05	0.6963875	0.05	2	0.069639
Item 2 b = +1	(2 - 1)	1	0.73	0.27				
Overall Likelihood (Sum of Products over All Thetas):								**0.228204**

(then multiply over all people)

(repeat with new values of item parameters until find highest overall likelihood)

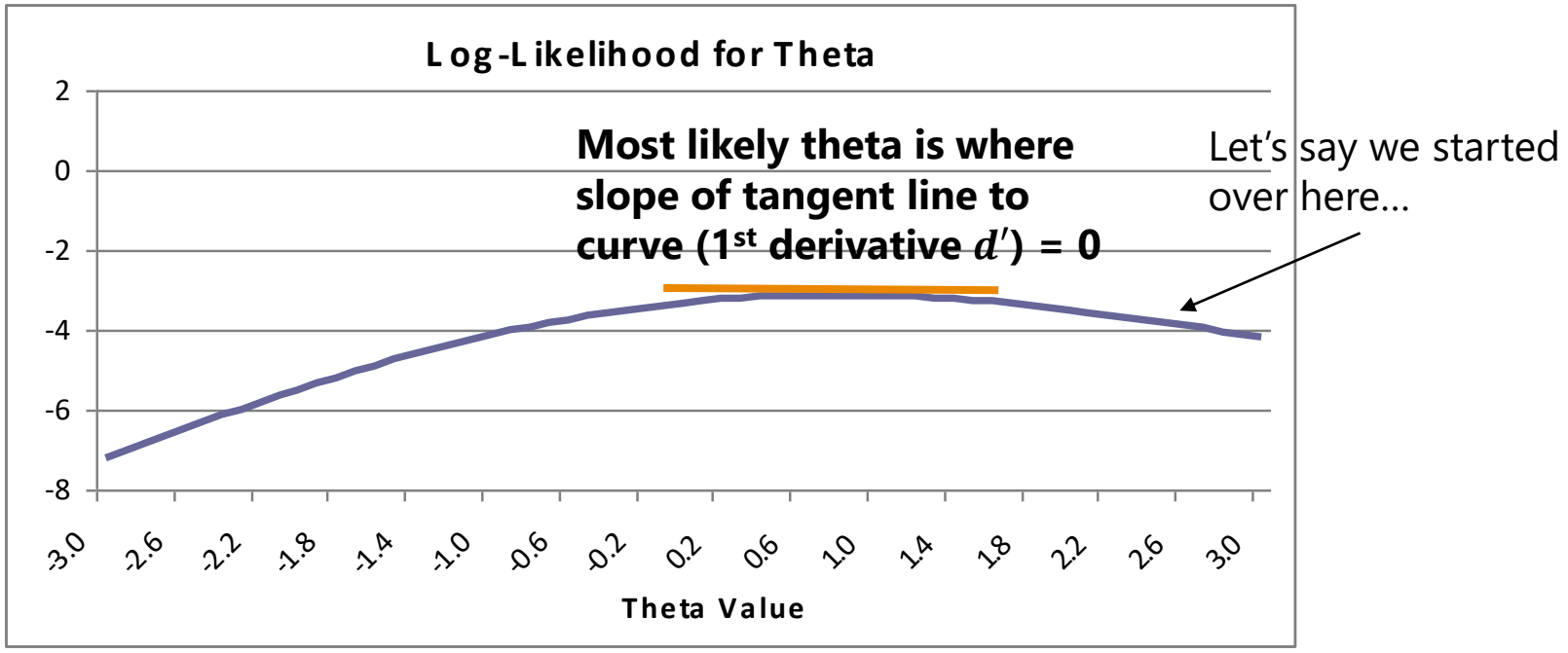
Once we have the item parameters, we can get some thetas…

- Let's say we are searching for theta given observed responses to 5 items with "calibrated" (known) difficulty values, so we try out 2 possible thetas
  - **Step 1**: Compute $\text{prob}(y_{is} = 1)$ using IRT model given each possible theta
    - $b_1 = -2, \theta_s = -1$: Logit$(y_{is} = 1) = (-1 - -2) = 1$, so $p(y_{is} = 1) = .73$
    - $b_5 = 2, \theta_s = -1$: Logit$(y_{is} = 1) = (-1 - 2) = -3$, so $p(y_{is} = 1) = .05$ → $p(y_{is} = 0) = .95$
  - **Step 2**: Multiple item probabilities together → product = "likelihood"
    - Products get really small, but if we take the log, then we can add them instead
  - **Step 3**: See which theta has the highest likelihood (here, $+2$)
    - More quadrature points → better estimate of theta
  - **Step 4**: Because subjects are independent, we can multiply all their response likelihoods together and solve all at once

Item	b	Y	Term	Value if…	
				θ = -1	θ = +2
1	-2	1	p	0.73	0.98
2	-1	1	p	0.50	0.95
3	0	1	p	0.27	0.88
4	1	1	p	0.12	0.73
5	2	0	1-p	0.95	0.50
Product of values:				0.01	0.30

Theta Estimation via Newton Raphson

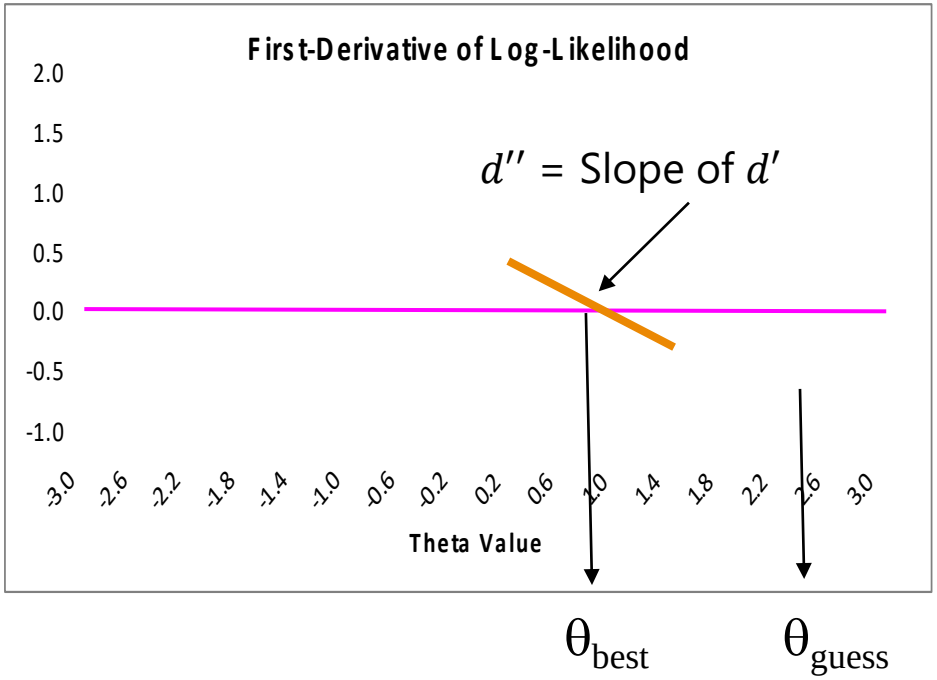
- We could calculate the likelihood over wide range of thetas for each subject and plot those likelihood values to see where the peak is…
  - But we have lives to lead, so we can solve it mathematically instead by finding where the slope of the likelihood function (the 1st derivative, d') = 0 (its peak)
- Step 1: Start with a guess of theta, **calculate 1st derivative d'** at that point
  - Are we there (d' = 0) yet? Positive d' = too low; negative d' = too high



Theta Estimation via Newton Raphson

- Step 2: **Calculate the 2nd derivative** (slope of slope, d'') at current theta guess
  - Tells us **how far off we are**, which is used to figure out how much to adjust by
  - d'' will always be negative as we approach top, but d' can be positive or negative
- Calculate new guess of theta: $\theta_{new} = \theta_{old} - (d'/d'')$
  - If $(d'/d'') < 0$ → theta increases
    If $(d'/d'') > 0$ → theta decreases
    If $(d'/d'') = 0$ then you are correct!

- **2nd derivative d'' also tells you how *good* of a peak you have**
  - Need to know where your best theta is (at $d' = 0$), as well as how precise it is (from d'')
  - If the function is flat, d'' will be smallish
  - **Want large d'' because 1/SQRT(d'') = theta's SE**



Theta Estimation: ML with Help

- ML is used to search for and find the most likely theta given observed item response pattern and the item parameters…
  …but can't estimate theta if item responses are all 0's or all 1's!

- **Prior distributions** to the rescue (yes, it's using Bayes)!
  - Multiply likelihood function for theta with prior distribution (usually we assume multivariate normal, as used in most software)
  - Contribution of the prior is minimized with increasing items, but allows us to get thetas for all 0 or all 1 response patterns

- Note the implication of this for what theta really is for each person:
  - **THETA IS A RANDOM EFFECT—A DISTRIBUTION, NOT A VALUE!**
  - Although we can find the most likely value, we can't ignore its probabilistic nature or how good of an estimate it is (how peaked the LL function is)
    - SE is constant across CFA factor scores, but SE is NOT constant across IRT/IFA thetas
  - **THIS IS WHY YOU SHOULD AVOID OUTPUTTING THETAS**

Factor/Theta Estimation: 3 Methods

- **ML**: Maximum Likelihood Scoring
  - Uses just the item parameters to predict the theta values
  - Can't estimate theta if none or all items are correct/endorsed
- **MAP**: Maximum a Posteriori Scoring
  - Combine ML estimate with a continuous normal prior distribution
  - Theta prediction is the **mode** of the posterior (prior+ML) distribution
  - Theta will be shrunken towards the mean if reliability is low
  - Is used in Mplus WLSMV (diagonally-weighted least squares, stay tuned)
- **EAP**: Expected A Posteriori Scoring
  - Combine ML estimate with a "rectangle" normal prior distribution
  - Theta prediction is the **mean** of the posterior (prior+ML) distribution
  - Is used in Mplus ML/MLR for CFA or IRT/IFA (and is best version)

What Goes Wrong for Absolute (Global) Model Fit using ML…

- **ML is a full-information estimator, and it is now trying to reproduce the observed item response pattern, <u>not a Pearson covariance matrix</u>!**
- Model DF is based on FULL response pattern: #responses$^{\#items}$
  - DF = # possible observed patterns – # parameters – 1
  - So, for an example of 24 binary items in a 1-P IRT model:
    - $\text{Max DF} = 2^{24} - \#a_i - \#b_i - 1 = 16{,}777{,}216 - 1 - 24 - 1 = \mathbf{16{,}777{,}190}$!
    - If some cells aren't observed (Mplus deletes them from the χ^2 calculation), then DF may be < Max DF, and thus χ^2 won't have the right distribution
- Pearson χ^2 based on classic formula: (observed – expected)^2 / expected
  - Good luck finding enough people to fill up all possible patterns!
  - Other χ^2 given in output is "Likelihood Ratio" χ^2, calculated differently
  - Linda Muthén suggests "if these don't match, they should not be used"
  - **χ^2 generally won't work well for assessing absolute global fit in IRT**

Local Model Fit Using ML IRT

- IRT programs (but not Mplus) provide "item fit" and "person fit" statistics
  - Item fit: Predicted vs. observed ICCs—how well do they match? Or via inferential tests (Bock Chi-Square Index or BILOG version)
  - Person fit "Z" based on predicted vs. observed response patterns
  - Many require the use of theta predictions, which makes them problematic!

- **Using ML in Mplus**: Local item fit available with **TECH10** output
  - **Univariate item fits**: How well did the model reproduce the observed response proportions? (Not likely to have problems here)
  - **Bivariate item fits**: Contingency tables for pairs of responses → Get χ^2 value for each pair of items for their remaining dependency after controlling for theta(s)

- Bivariate item fit is the basis of the newest absolute fit statistics (e.g., work by Maydeu-Olivares): M_2 (analogous to χ^2 test), RMSEA_2, and SRMR_2
  - The M_2 statistic indexes global fit by computing a χ^2(observed vs. expected metric) for the fit to the marginal frequency of each item's responses and two-way cross-tabulations for each pair of item responses
    - Not currently in Mplus, but available as M2 function within the MIRT package in R

Summary: ML Estimation for IRT Models

- Full-information Marginal ML with numeric integration for IRT models tries to find the item parameters that are most likely *given the observed item response pattern* → IFA or IRT parameters using logit or probit scales

- Because of the integration (i.e., rectangling of theta) required at each step of estimation, it may not be feasible to use ML for IRT models in small samples or for many factors at once (too many rectangles simultaneously)
  - This where MCMC (Bayesian) estimation can be a more practical strategy!

- IRT using ML does not have agreed-upon measures of absolute global fit
  - Categorical item responses cannot be summarized by just a Pearson covariance matrix, but by all possible response patterns (full contingency table) instead
  - Usually there are not enough people to fill up all possible response patterns, so there's no valid basis for an absolute fit comparison using "expected"
  - Nested models (on <u>same</u> items!) can still have relative fit compared via $-2\Delta LL$

- There is another game in town for IRT/IFA estimation in Mplus, however…

Another Alternative: WLSMV

- **WLSMV:** "Weighted Least Square parameter estimates use a diagonal weight matrix and a Mean- and Variance-adjusted χ^2 test"
  - Called "diagonally-weighted least squares" (DWLS) by non-Mplus people

- Translation: **WLSMV** is a **limited-information** estimator that uses a different summary of responses instead → **a "linked" covariance matrix**

- Fit can then be assessed in regular CFA ways, because what is trying to be reproduced is again a **type of covariance matrix**
  - Instead of the *full item cross-tabulation of response patterns* (as in ML)
  - We can then get the typical measures of absolute fit available in CFA

- Normally CFA uses the *Pearson* covariance matrix of the items…
  - But correlations among binary items will be less than 1 any time p differs from .5, so the covariances will be restricted as well…
  - What if we could fit a covariance matrix on the logit or probits instead???

Bivariate Association of Binary Variables

- The possible **Pearson's r for binary variables will be limited** when they are not evenly split into 0/1 because their variance depends on their mean
  - Remember: Mean = p_i , Variance = $p_i(1-p_i) = p_i q_i$

- If two binary variables (x_i and y_i) differ in p_i, such that $p_y > p_x$
  - Maximum covariance: $Cov(x,y) = p_x(1-p_y)$
  - This problem is known as **"range restriction"**
  - **Here this means the maximum Pearson's r will be smaller than ± 1 it should be:**

$$r_{x,y} = \sqrt{\frac{p_x(1-p_y)}{p_y(1-p_x)}}$$

  - Some examples using this formula to predict maximum Pearson r values →
  - **So Pearson correlations may not adequately describe relations of categorical variables…**

px	py		max r
0.1	0.2		0.67
0.1	0.5		0.33
0.1	0.8		0.17
0.5	0.6		0.82
0.5	0.7		0.65
0.5	0.9		0.33
0.6	0.7		0.80
0.6	0.8		0.61
0.6	0.9		0.41
0.7	0.8		0.76
0.7	0.9		0.51
0.8	0.9		0.67

Correlations for Binary or Ordinal Variables

- **Pearson correlation**: between two quantitative variables, working with the observed distributions as they actually are
- **Phi correlation**: between two binary variables, still working with the observed distributions (= Pearson with computational shortcut)
- **Point-biserial correlation**: between one binary and one quantitative variable, still working with the observed distributions (and still = Pearson)

Line of Suspended Disbelief to Reduce Impact of Range Restriction

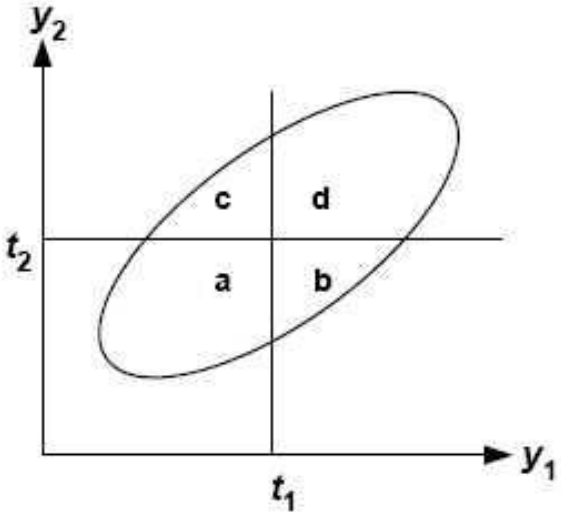
- **Tetrachoric correlation**: between "underlying continuous" distributions of two actually binary variables (not = Pearson) → based on probit!
- **Polychoric correlation**: between "underlying continuous" distributions of two ordinal variables (not = Pearson) → based on probit!
- **(Bi/Poly)serial correlation**: between "underlying continuous" (but really binary/ordinal) and observed quantitative variables (not = Pearson)

- Bivariate statistics related to categorical variables should be provided using (**tetra/poly)choric or (bi/poly)serial correlations** instead of Pearson

Tetrachoric and Polychoric Correlation

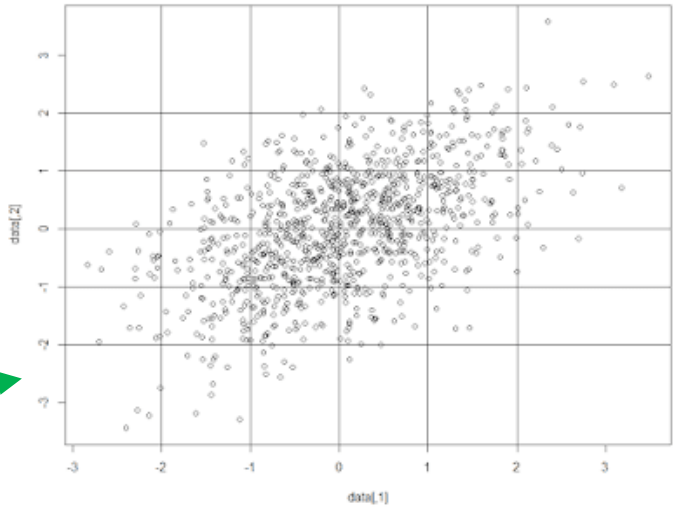
Data	$y_2 = 0$	$y_2 = 1$
$y_1 = 0$	a	c
$y_1 = 1$	b	d

Tetrachoric reasoning: Given a bivariate normal distribution of the underlying continuous variables (y^* version), what correlation would have created the observed proportion in each quadrant (→ cell)?



- Polychoric and tetrachoric correlations are similar:
  - Both based on a bivariate normal distribution,
  - Both try to represent the correlation that would have created the proportion of responses in each cell (unique combo of row by column)

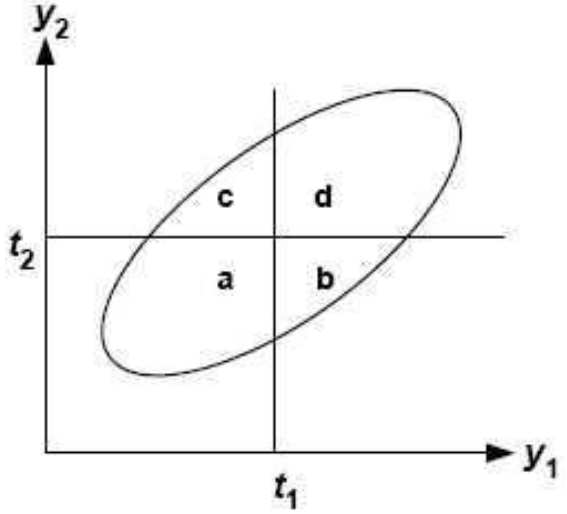
- See this website for a more thorough description with this helpful example of the extension to polychoric!



WLSMV Estimation (Diagonally Weighted Least Squares)

Data	$y_2 = 0$	$y_2 = 1$
$y_1 = 0$	a	c
$y_1 = 1$	b	d

Tetrachoric reasoning: Given a bivariate normal distribution of the underlying continuous variables (y^* version), what correlation would have created the observed proportion in each quadrant (→ cell)?



- WLSMV first estimates correlation matrix of underlying item responses (probit scale)
  - For **binary** responses → **tetrachoric** correlation matrix as new H_1 **saturated** model
  - For **ordinal** (polytomous) responses → **polychoric** correlation matrix as H_1 **saturated** model
- The model then tries to find item parameters to predict this new correlation matrix
- The diagonal W "weight" part then tries to emphasize reproducing underlying variable correlations that are relatively well-determined more than those that aren't
  - The full weight matrix is of order z*z, where z is number of matrix elements to estimate
  - The "diagonal" part means it only uses the *preciseness of the estimates themselves*, not the covariances among the "preciseness-es" (much easier, and not a whole lot of info lost)
- The "MV" corrects the χ^2 test for bias arising from this weighting process

More about WLSMV Estimation

- Works much faster than ML when you have small samples or many factors to estimate (because no rectangling via quadrature is required)
- **Does assume missing data are missing completely at random,** whereas ML assumes only *missing at random* (conditionally random)
- Because a saturated covariance matrix is used as the input data, we get **absolute fit indices** as in CFA, but they should be interpreted with caution
  - Fewer people → less well-estimated "saturated" matrix to start from
  - More skewness, fewer categories → easier to get falsely good model fit
- Model parameters must be on the **probit scale** instead of logit scale
  - Unlike full-information ML, in which you can choose logit or probit instead
- Two item variance scalings in Mplus via the **PARAMETERIZATION** option on the ANALYSIS command, where a 1 is needed for identification
  - "**Delta**" (default): Var(y_i^*) = factor + error = 1 = "marginal parameterization"
  - "**Theta**": Var(e_i^*) = 1 instead = "conditional parameterization"
    - **WE WILL USE THIS ONE TO HELP SIMPLIFY IRT CONVERSIONS**

Model Comparisons with WLSMV using DIFFTEST in Mplus

- Not the same process! Model DF is NOT calculated in usual way, and model fit is not compared in the usual way
  - Absolute χ^2 model fit values are meaningless—they are not comparable!
  - Difference in model χ^2 are not distributed as χ^2 anymore

- Here's how you do nested model comparisons in WLSMV:
  - Step 1: Estimate model with *more* parameters, adding this command:
    - SAVEDATA: DIFFTEST=more.dat; → Saves needed derivatives to file
  - Step 2: Estimate model with *fewer* parameters, adding this command:
    - ANALYSIS: DIFFTEST=fewer.dat; → Uses those derivatives to do $\Delta\chi^2$ test
  - Step 2 model output will have a new χ^2 difference test in it that you can use, with DF difference to compare to a χ^2 distribution

Assessing Local Model Fit

- **Need to check local model fit is the same in IRT/IFA as in CFA**

- **Using ML**: Local item fit in Mplus available with **TECH10** option
  - **Univariate item fits**: How well did the model reproduce the observed response frequencies? (Not likely to have problems here if each item has own location)
  - **Bivariate item fits**: Contingency tables for pairs of responses → Get χ^2 value for each pair of items for their remaining dependency after controlling for Theta(s)
    - Done for every pair of items, so there will be LOTS of output to wade through ☹

- **Under WLSMV**: Residual correlation matrix (i.e., model–data discrepancy) via the RESIDUAL option on OUTPUT statement (just like in CFA)
  - Predicted and residual (discrepancy) item tetrachoric/polychoric correlations
  - Look for large (absolute value) discrepancies in correlation metric
  - Will be MUCH easier to do for many items than all bivariate item fits in ML!

Error Covariances in IRT/IFA

- Additional relationships between items can be included:
  - Via **error covariances** (the same as in CFA) when using **WLSMV** because the model is being estimated on the tetrachoric/polychoric correlation matrix (so the error of the underlying probit can covary, even if item error or total variances =1 for identification)
  - Error covariances are not allowed when using full-information ML
  - Instead, you can specify "**method factors**" (in WLSMV or ML), also known as a "**bifactor model**" (which can also be used in CFA models)
- Here is an example using WLSMV to demonstrate both ways:

```
! Primary factor/theta
  Trait BY item1-item5*;
  [Trait@0]; Trait@1;
! Error covariance
  item2 WITH item3*;
```

```
! Primary factor/theta
  Trait BY item1-item5*;
  [Trait@0]; Trait@1;
! Uncorrelated factor to
  create error covariance
  ErrFact BY item2@1 item3@1;
 [ErrFact@0]; ErrFact*;
  ErrFact WITH Trait@0;
```

Error Covariances in IRT/IFA

```
! Primary factor/theta
  Trait BY item1-item5*;
  [Trait@0]; Trait@1;
! Uncorrelated factor to
  create error covariance
  ErrFact BY item2@1 item3@1;
 [ErrFact@0]; ErrFact*;
  ErrFact WITH Trait@0;
```

For models with many method factors, add the **ANALYSIS:** option **MODEL=NOCOVARIANCES** to made all factors **uncorrelated** by default (instead of all factors correlated by default as usual)

```
TRAIT    BY
    ITEM1     0.994      0.078     12.724      0.000
    ITEM2     2.138      0.148     14.459      0.000
    ITEM3     1.823      0.125     14.527      0.000
    ITEM4     1.106      0.090     12.311      0.000
    ITEM5     0.232      0.045      5.200      0.000

ERRFACT  BY
    ITEM2     1.000      0.000    999.000    999.000
    ITEM3     1.000      0.000    999.000    999.000

ERRFACT  WITH
    TRAIT     0.000      0.000    999.000    999.000

Variances
    TRAIT     1.000      0.000    999.000    999.000
    ERRFACT   1.996      0.314      6.357      0.000
```

To create a negative error covariance, fix the ErrFact loadings to 1 and –1 instead.

The variance of ErrFact then predicts a positive additional covariance for item 2 with item 3.

IRT/IFA Model Estimation: Summary

- Full-information Marginal ML estimation with numeric integration provides:
  - **"Best guess"** as to the value of each item parameter (and person theta if you ask for it)
  - **SE** that conveys the uncertainty of that prediction

- The "**best guesses"** for the model parameters do not depend on the sample:
  - Item estimates do not depend on the particular individuals that provided responses
  - Person estimates do not depend on the particular items that were administered
  - Thus, model parameter estimates are sample-invariant

- The **SEs** for those model parameters DO depend on the sample
  - Item parameters will be estimated less precisely **where** there are fewer individuals
  - Person parameters will be estimated less precisely **where** there are fewer items

- **WLSMV (DWLS)** in Mplus uses limited-information estimation for IFA or IRT models
  - Uses an estimated tetrachoric correlation matrix as input for the factor analysis
  - Works better for many factors than ML (but can be less trustworthy overall)
  - But beware of missing data! ML assumes MAR, whereas WLSMV assumes MCAR instead!