On Test Scores

On Test Scores

Multidimensional Measurement Models, Fall 2023

Today’s Class

- Scores
  - Types of scores
    - Sum scores / test scores
    - Factor scores
  - Score contents
  - Relating sum scores to factor scores
  - Score reliability
- Why using scores alone in separate analysis, while done almost always, is not good practice

The Big Picture

- Ideally, latent multivariate analyses with (and without) measurement error should be conducted simultaneously
  - Error propagates

- There are many instances when one cannot do a simultaneous analysis
  - This lecture is an attempt to get you *as close to* results from a simultaneous analysis by getting you to understand the psychometric and statistical properties of using scores

What’s In a Sum Score?

The Purpose of this Lecture: Some Clarity on Scores

- As I’ve been a student and a teacher I have found the topic of scores to be incomplete and often contradictory

- Some things I’ve heard:
  - “Sum scores are almost always okay”
  - “Factor scores (think GRE) are okay if they are from some strange sounding model…”
  - “…otherwise factor scores are the work of the devil”

The Purpose of this Lecture: Some Clarity on Scores

- A question that I hearing: Why use Structural Equation Modeling (or CFA/IRT) when I can just use a sum of the items?
  - Sum of the items == sum score == total score == Add stuff up (ASU) model

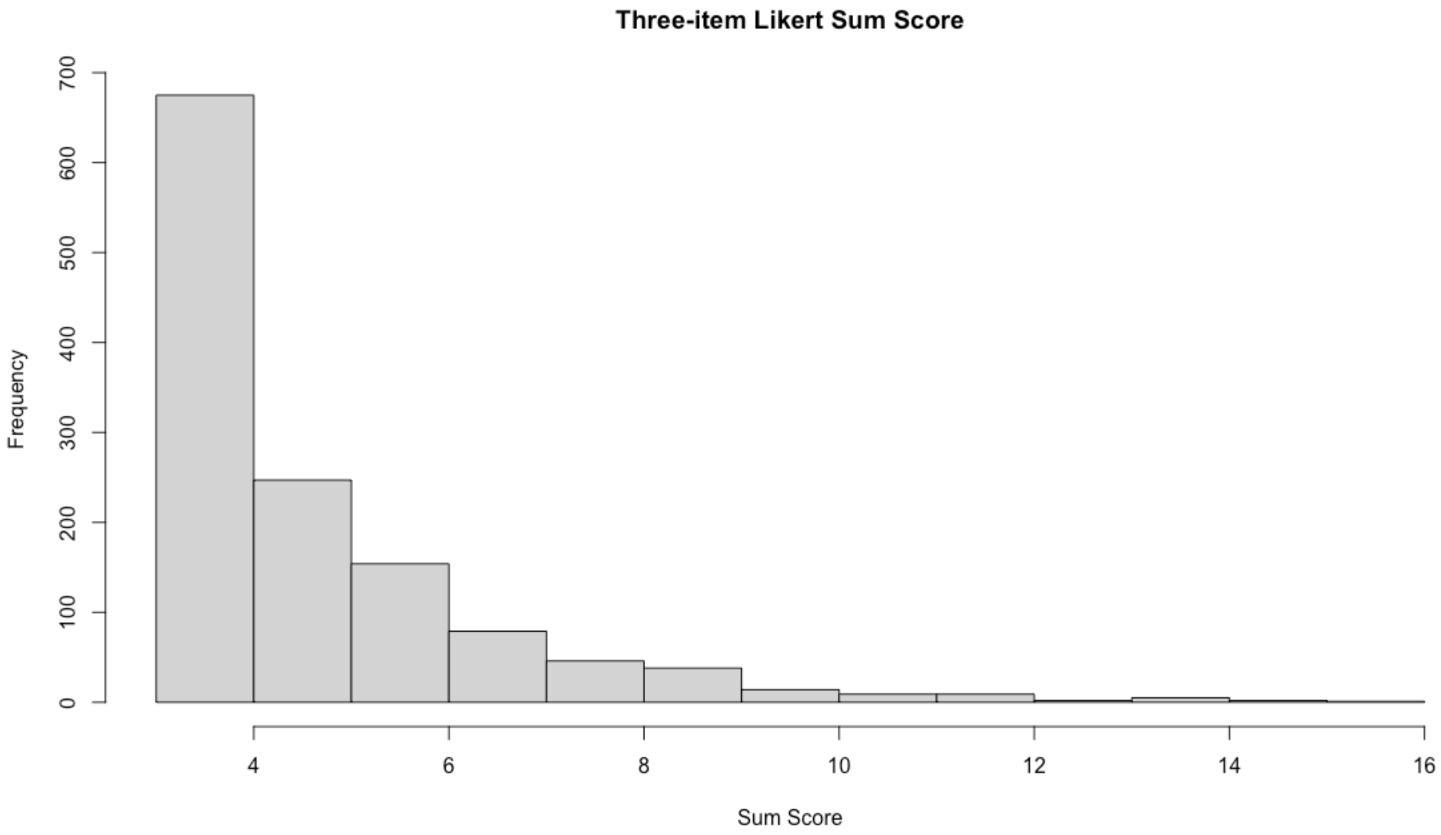
- Sum score are used as:
  - Observed variables in secondary analyses
  - Results given to participants, patients, students, etc…

- Current practice in educational research seems to be:
  - Use a sum score until some reviewer says you cannot use one
  - At that point, use a confirmatory factor model to verify that you have a one-factor scale
  - …then use a sum score

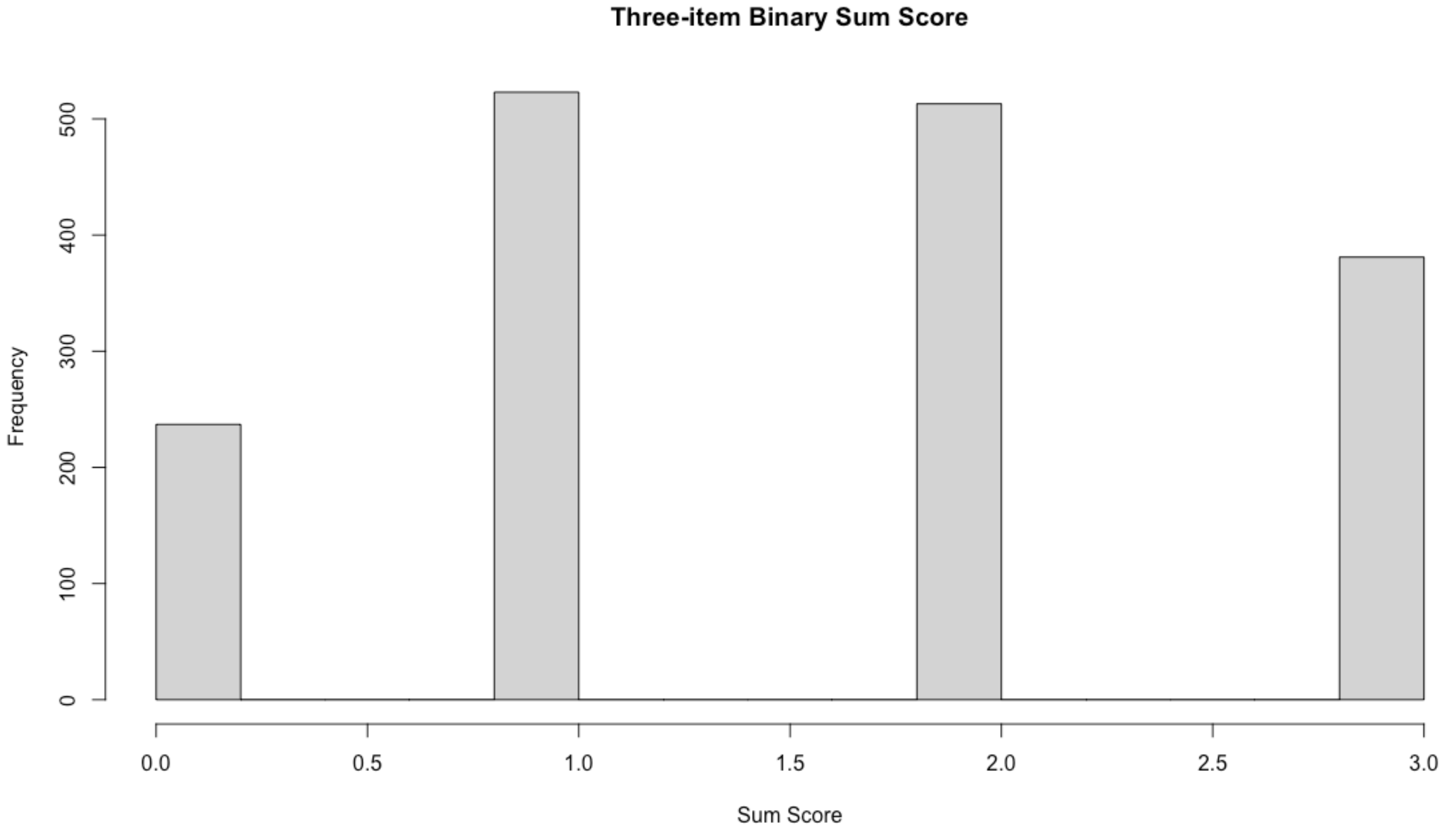
Demonstration Data

- To demonstrate the concepts appearing throughout this section, we will use two three-item data sets:
  - A three-item Likert-type scale used for measuring pathological gambling (Items: GRI1, GRI3, and GRI 5)
    - Each item had six integer-valued categories (from 1 to 6)
  - A three binary items from my example data
    - Called binaryData
    - Three ELA reading items

Distribution of Likert Sum Scores



Distribution of Binary Sum Scores



Psychometric Properties of Sum Scores

- The use of sum scores brings about a discussion about the psychometrics that underlie sum scores

- What you have learned about measurement so far likely falls under the category of CTT:
  - Writing items and building scales
  - Item analysis
  - Score interpretation
  - Evaluating reliability and construct validity

- Big picture: We will view CTT as model with a restrictive set of assumptions within a more general family of latent trait measurement models
  - Confirmatory Factor Analysis is a measurement model

Differences Among Measurement Models

- What is the **name of the latent trait** measured by a test?
  - Classical Test Theory (CTT) = "True Score" (T)
  - Confirmatory Factor Analysis (CFA) = "Factor Score" (F)
  - Item Response Theory (IRT) = "Theta" (θ)

- Fundamental difference in approach:
  - **CTT → unit of analysis is the WHOLE TEST** (item sum or mean)
    - **Sum = latent trait**, and the sum doesn't care how it was created
    - Only using the sum requires restrictive assumptions about the items
  - **CFA, IRT, <u>and beyond</u> → unit of analysis is the ITEM**
    - Model of how item response relates to an **estimated latent trait**
    - Different models for differing item response formats
    - Provides a framework for testing adequacy of measurement models

Classical Test Theory: Assumed Model

- In CTT, the TEST is the unit of analysis:

$$Y_{\text{Total}} = T + e$$

  - **True score T:**
    - Best estimate of 'latent trait': Mean over infinite replications
    - Scale of T is the same as the scale of Y_{Total}
  - **Error e:**
    - Expected value (mean) of 0, expected to be uncorrelated with T
    - Supposed to wash out over repeated observations

- **So the expected value of $\boldsymbol{Y_{total}}$ is $\boldsymbol{T}$**
  - Put another way: should the model fit, Y_{total} is an unbiased estimate of T
  - The true score is why you created the sum in the first place→ your test purports to measure one thing, bringing about one sum score per person

Classical Test Theory: Assumed Model

- No distributional assumptions made…yet

- Even if your data fit a one-factor model, when using a sum score, the error portion is part of Y_{Total}
  - But, it is only one part of the error that is in a sum score

- Because the CTT model does not include individual items, items must be assumed exchangeable
  - If the model fits, then more items means better reliability

More CTT Basics

- A goal of CTT is to quantify reliability
  - Reliability is the proportion of variance in the sum score that is due to variation in the latent trait

- Reliability decomposition comes from Var(Y)
  - Var() function comes from the expected value in mathematical statistics
  - $E\left(g(x)\right) = \int g(x)f(x)dx$
    - Over the sample space/support of x with probability density function f(x)
    - Replace integral with a sum for discrete x (and pdf for probability mass function)
  - Mean: $\mu = E(x) = \int x\, f(x)dx$
  - Variance: $Var(x) = E\left((x-\mu)^2\right) = E\left[\left(x - E(x)\right)^2\right] = \int (x-\mu)^2\, f(x)dx$

More CTT Basics

- For CTT:

$$\mathrm{Var}(Y_{Total}) = \mathrm{Var}(T+e) = \mathrm{Var}(T) + Var(e) + 2Cov(T,e)$$

- But, $Cov(T,e) = 0$ as T and e are assumed independent, so

$$\mathrm{Var}(Y_{Total}) = \mathrm{Var}(T) + Var(e)$$

Moving from Variance to Reliability

- Reliability, as a proportion of variance in sum score due to the trait:

$$\rho = \frac{\text{Var}(T)}{\text{Var}(Y)} = \frac{\text{Var}(T)}{\text{Var}(T) + \text{Var}(e)}$$

  - $\text{Var}(Y)$ == variance of observed sum score
  - $\text{Var}(T)$ == variance of true score == variability in the unobserved latent trait == **individual differences**
  - $\text{Var}(e)$ == variance of error == **measurement error**
- Key question: how does one quantify reliability?
  - We will see that depends….

Draw Templin, Draw!

- Picture of distributions of Y, T, and e…

Parceling: Creating Another Type of Sum Score

- Another type of sum score is a parcel (sometimes called an item parcel or an item bundle)
  - A parcel then takes the places of the summed variables in a larger structural equation model
- Today, I make the case that any sum of items (or any quantities, really) makes strict, testable measurement model assumptions
  - A parcel assumes a CTT model: $Y_{\text{Total}} = T + e$
- Therefore, properties of CTT scores applies to parcels
- Another note: Parceling is frequently hides model misfit
  - But…we can test whether the assumptions of a sum score are met when parceling using a CFA-like framework

Potential Sources of Error Sum Scores

- Measurement error
  - e.g., the e in $Y = T + e$

- Model misspecification error of various types:
  - Dimensionality misspecification error
    - e.g., Assuming one dimension when there is more than one present
  - Parameter constraint misspecification error
    - e.g., Assuming overly restrictive constraints (see next section and all of CTT)
  - Linear model functional misspecification error
    - e.g., Assuming a linear relationship between the factor and the items when a non-linear one is present
  - Outcome distribution misspecification error
    - e.g., Assuming Likert-type data to be continuous and using a normal distribution
  - Factor distribution misspecification error
    - e.g., Assuming your trait is normally distributed when it is categorical or a mixture distribution

- Missing data error
  - How you treat missing responses to items makes even more untenable assumptions

- ~~Sampling error~~
  - (meaning error in parameters due to small n) is **not a source of error** in a sum score
  - Note: measurement error is sampling error with respect to items instead of people

Why Error Matters

- Ignoring error will lead to inaccurate and potentially misleading results
  - Biased estimates (Type II error)
  - Biased standard errors of estimates (Type I error)
- Some sources of error matter more than others
- Measurement error is often thought of as the worst, but I believe model misspecification error to be even worse than measurement error

Factor Scores

Factor Scores

- To describe a factor score, first remember the CFA model:

$$Y_{pi} = \mu_i + \lambda_i F_p + e_{pi}$$

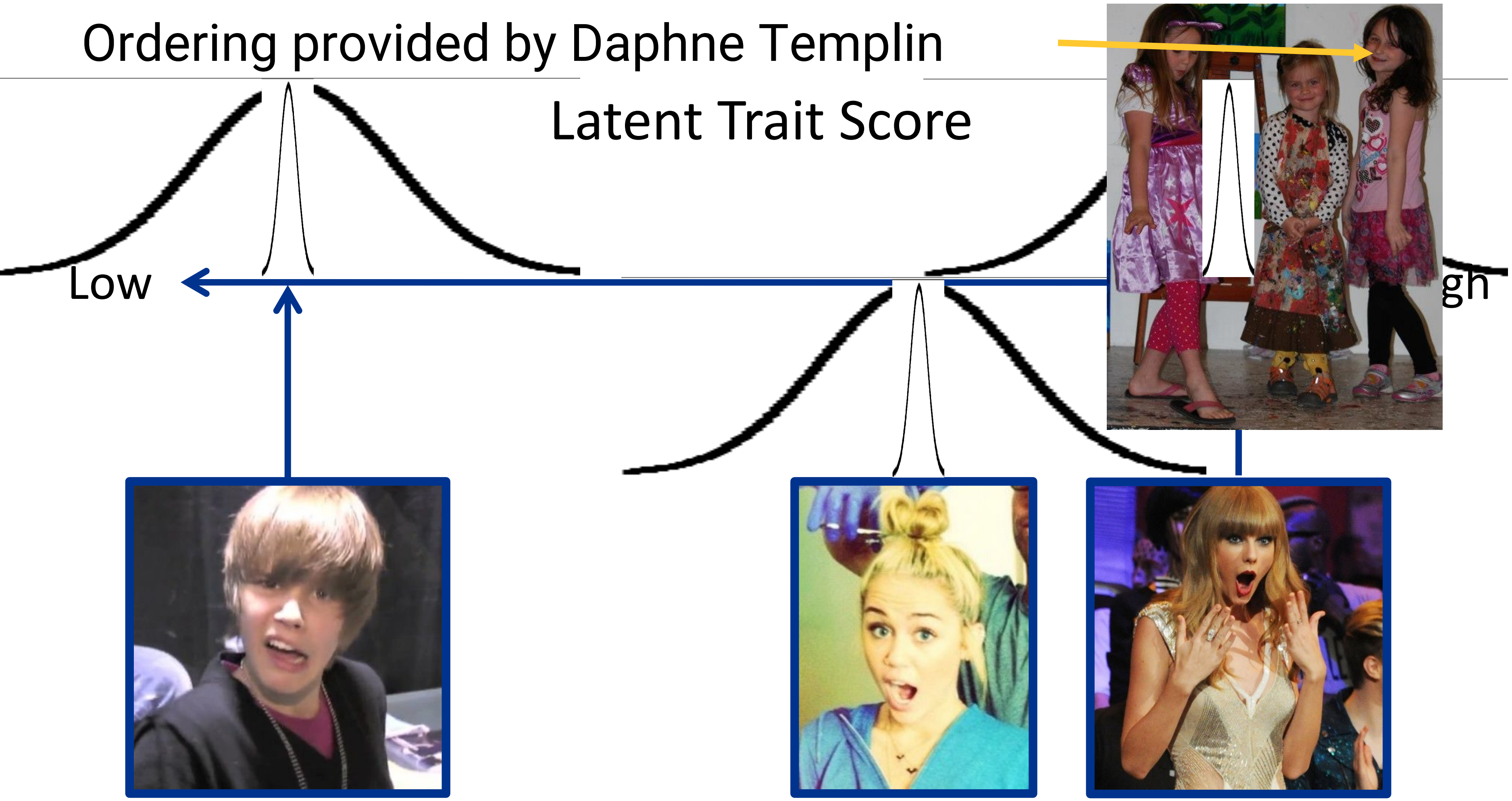
- Simply put: A factor score is an estimated value for F_p, or $\hat{F}_p$

- There has long been a resistance to using factor scores in psychological research with the most common objection cited being the **indeterminacy of factor scores**
  - **Indeterminacy of factor scores == factor scores are not unique**

- Why are factor scores not unique? Because factor models must fix some parameters for identification
  - The values may be indeterminate—but in CFA and in ML versions of EFA **the rank order of the factor scores is unique**

- There is also rotational indeterminacy:
  - More than one solution of factors will fit equally well

A Depiction of Traditional Psychometric Estimates



Ordering provided by Daphne Templin

Latent Trait Score

Low

gh

Bieber

Cyrus

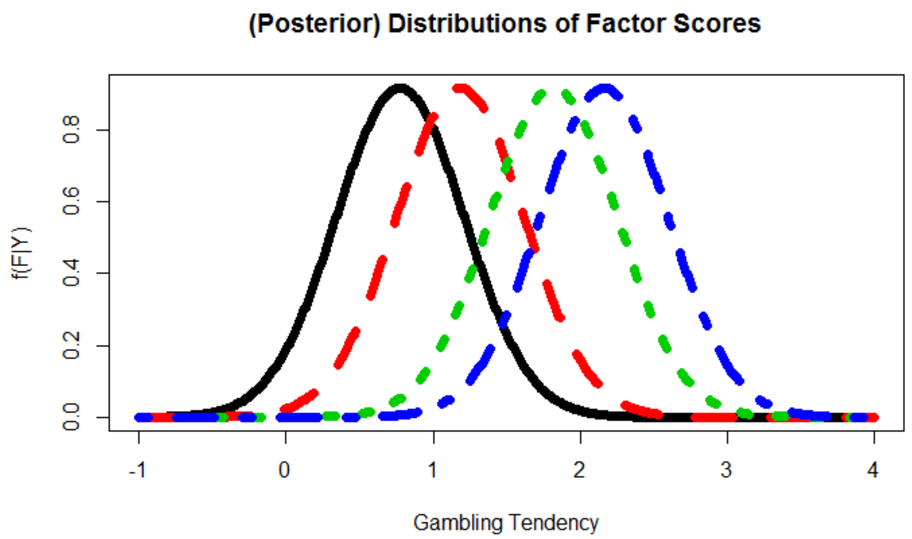
Swift

Factor scores provide a **weak ordering** of people
(weak because of error): like ordinal-level measurement

Example Factor Score Distributions

- Factor scores provide a **weak ordering** of people ("weak" because of error)

- A different version of factor model identification would change the numbers on the X-axis, but the shapes and order of the distributions would not change

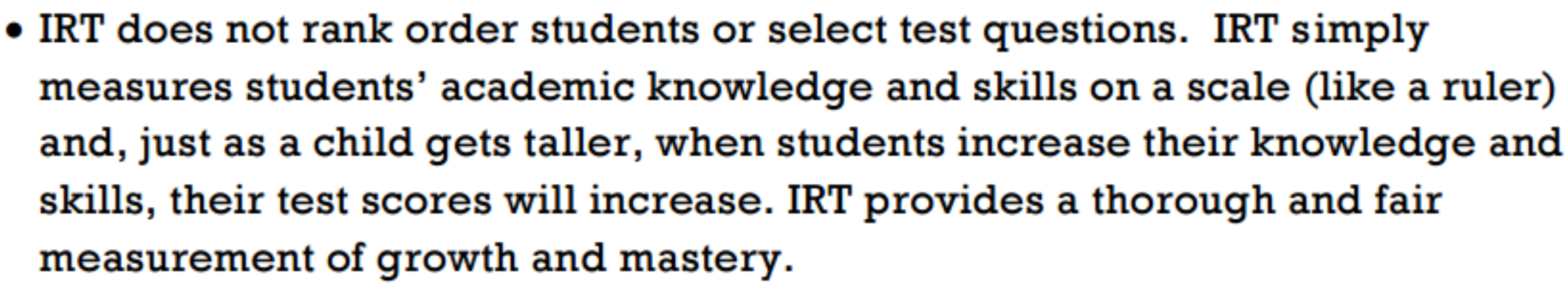


Factor Scores and Testing

- These factor scores are found using the same methods as are used in practice for finding test scores (like the GRE)
  - The only difference between such test scores and factor scores is the distributional assumptions of the measurement model (IRT is CFA with assumed Bernoulli/Multinomial distributed items)
  - They behave the same

- That said, some in the testing industry don't quite realize how these work

From 2015: http://images.pearsonassessments.com/images/tmrs/Responses_Walter_Stroup.pdf (p. 2)



- IRT does not rank order students or select test questions. IRT simply measures students' academic knowledge and skills on a scale (like a ruler) and, just as a child gets taller, when students increase their knowledge and skills, their test scores will increase. IRT provides a thorough and fair measurement of growth and mastery.

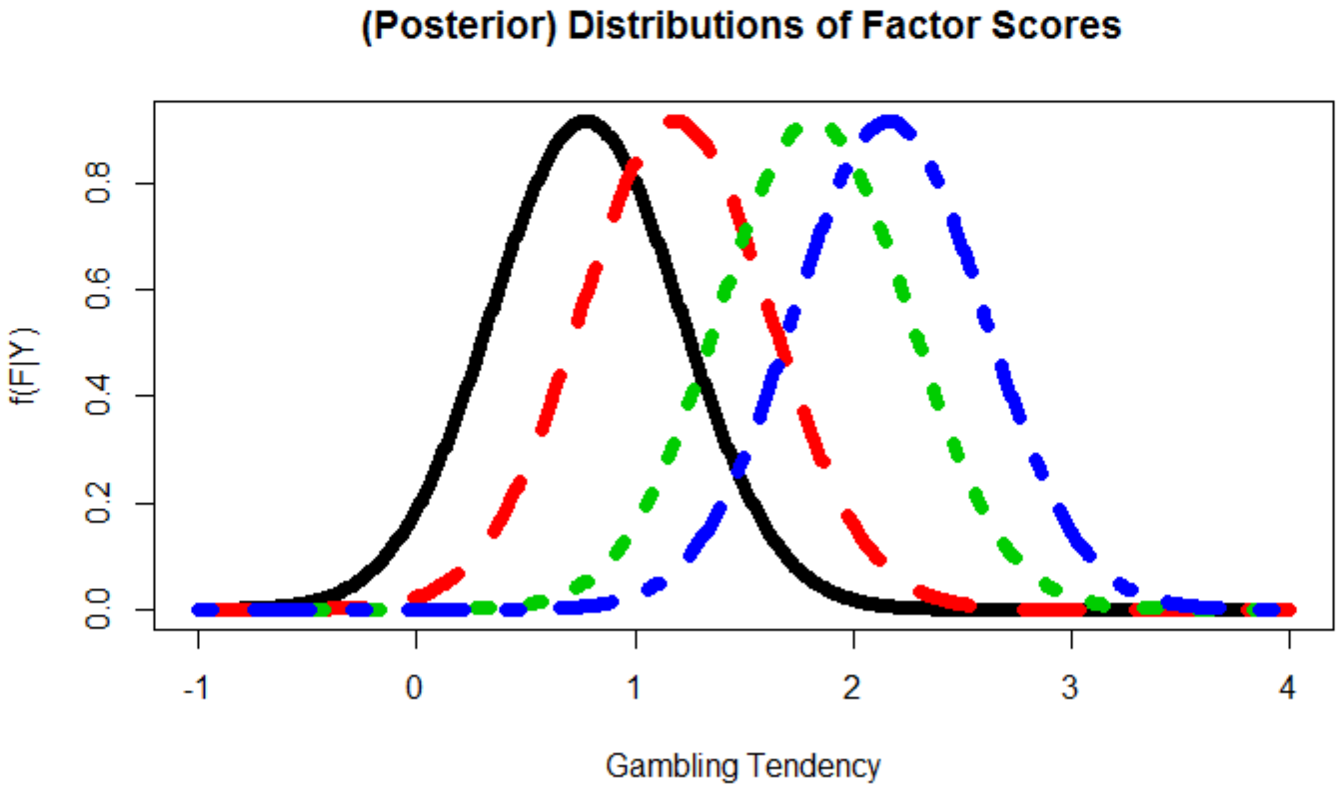
More on Factor Scores

- Factor scores (by other names) are used in many domains
  - Item response theory (CFA with categorical items): GRE scores are factor scores
- Because the historical relationship between CFA and exploratory factor analysis, factor scores are widely avoided
  - In EFA factor meaning is unknown so rotations were used
- Further making the issue even more difficult, many methods for determining factor scores have been developed
  - See http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3773873/
- We will only focus on one method for estimating factor scores that is used in nearly all fields based on the posterior distribution of the factor score given the data
  - Identical to methods described by Lawley and Maxwell (1971) of Bartlett (1936)
  - Also used in generalized linear mixed effects models where factor scores are called Best Linear Unbiased Predictors (or BLUPs)

Factor Scores: The Big Picture

- A factor score is the estimate of a subject’s unobserved latent trait
- Because this latent variable is not measured directly, it acts like it is missing data: you really cannot know with certainty its true value
- It is difficult to pin down what the missing data value (factor score value) should be precisely
  - Each factor score has a posterior distribution of possible values
  - Often, the mean of the posterior distribution is the “factor score”
    - In CFA, the mean is the most likely value
  - Depending on the test, there may be a lot of error (variability) in the distribution
- Therefore, the use of factor scores must reflect that the score is not known and is represented by a distribution

Factor Score Distributions



How Distributions get Summarized into Scores

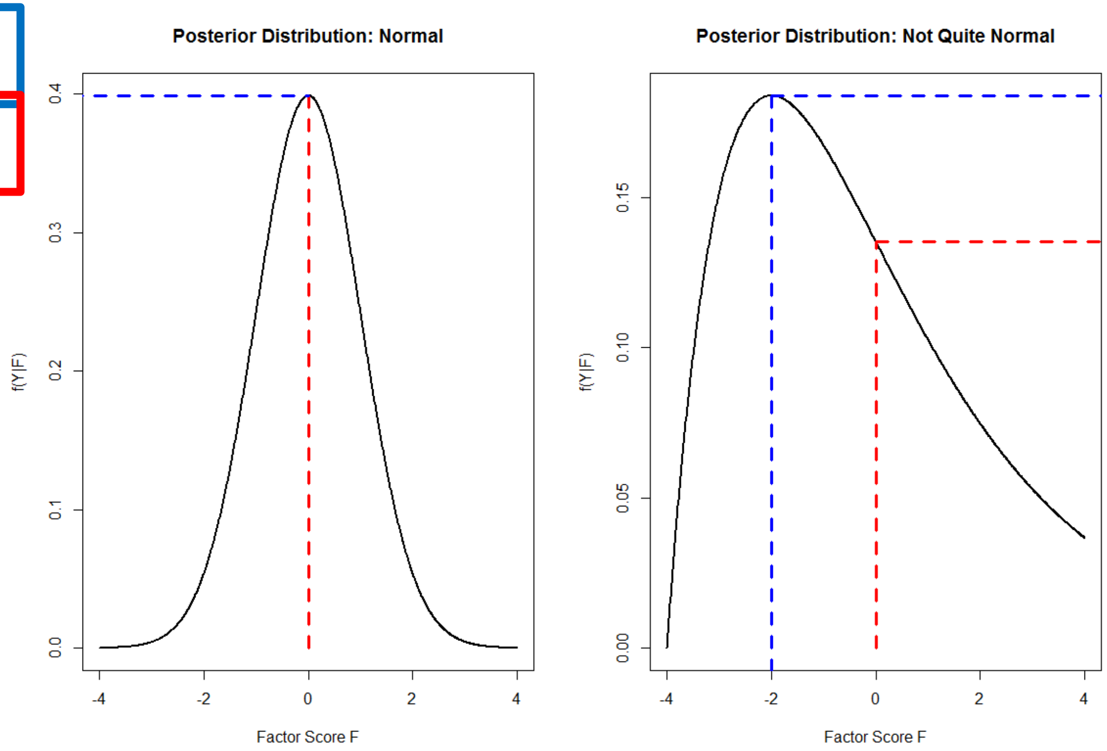
- There are two ways of providing a score from the factor score posterior distribution:
  - Expected a posteriori (EAP): the mean of the distribution
  - Maximum a posteriori (MAP): the most likely score from the distribution

- In CFA factor score distributions are normal (so EAP=MAP)
  - Derivation forthcoming

Types of Factor Scores

MAP

EAP



MAP

EAP

Additional Information on Factor Scores

- For EAP factor scores:
  - $\hat{F}_p = E\left(f\left(F_p|\mathbf{Y}\right)\right)$
  - $SE\left(\hat{F}_p\right) = \sqrt{Var\left(f\left(F_p|\mathbf{Y}\right)\right)}$
- For MAP factor scores:
  - $\hat{F}_p = \arg\max_{F_p} f\left(F_p|\mathbf{Y}\right)$
  - $SE\left(\hat{F}_p\right) = \left[\frac{\partial^2}{\partial F_p^2} f\left(F_p|\mathbf{Y}\right)\Big|_{\hat{F}_p}\right]^{-\frac{1}{2}}$ (square root of Fisher's information)
- For CFA (Normal Data/Normal Factor) measurement models:
  - MAP = EAP
  - Variance is identical across all people, regardless of score
- For non-CFA measurement models:
  - MAP $\neq$ EAP (but does with infinite items)
  - Standard error is a function of the factor score

Tying Factor Scores to Classical Test Theory

- Recall Classical Test Theory's model:

$$Y = T + E$$

- With reliability: $\rho = \frac{Var(T)}{Var(T)+Var(E)}$

- For factor scores:
  - $Var(T) = \sigma_F^2$: the (possibly estimated) variance of the factor
  - $Var(E) = SE\left(\hat{F}_p\right)^2$: From the posterior distribution of the factor score

- Therefore, reliability of factor scores can be computed using model-estimated parameters
  - Caution: The factor model must fit to use these parameters!
  - Caveat: We'll soon see reliability for sum scores can be estimated by CFA model parameters

Factor Scores:
Empirical Bayes Estimates

- **For most latent variable techniques**, the factor scores come from Empirical Bayes estimation—meaning there is a prior distribution present
  - Empirical = some or all of the parameters of the distribution of the latent variable are estimated (i.e., factor mean and variance)
  - Bayes = comes from the use of Bayes' Theorem

- Prior == Assumed factor distribution with mean/variance

- This is true for all CFA, IRT, mixed/multilevel/hierarchical models
  - And is true for models that don't have a label (e.g., Poisson Factor Analysis?)

- Note: So-called ML-based factor scores are frequently estimated
  - These are analogous to the MAP scores from Bayesian methods, just without a prior distribution

Bayes' Theorem

- Bayes' Theorem states the conditional distribution of a variable A (soon to be our factor score) given values of a variable B (soon to be our data) is:

For Categorical A, replace integral with sum

$$f(A|B) = \frac{f(B|A)f(A)}{f(B)} = \frac{f(B|A)f(A)}{\int_{a \in A} f(B|A=a)f(A=a)da}$$

- $f(A|B)$ is the **distribution** of A, conditional on B
  - We will come to know this as the posterior distribution of the factor score, conditional on the data observed or $f(\mathbf{F}|\mathbf{Y})$
- $f(B|A)$ is the **distribution** of B, conditional on A
  - We will come to know this as our measurement model or $f(\mathbf{Y}|\mathbf{F})$
- $f(A)$ is the marginal distribution of A
  - We will come to know this as the prior distribution of the factor or $f(\mathbf{F})$

Putting Together the Pieces of Empirical Bayes Factor Scores

$$f(A|B) = \frac{f(B|A)f(A)}{f(B)} = f(\mathbf{F}|\mathbf{Y}) = \frac{f(\mathbf{Y}|\mathbf{F})f(\mathbf{F})}{f(\mathbf{Y})}$$

- For $f(\mathbf{Y}|\mathbf{F})$, consider the measurement model (here CFA) for one item:

$$Y_{pi} = \mu_i + \lambda_i F_p + e_{pi}$$

Where: $e_{pi} \sim N\left(0, \psi_i^2\right)$

- Using expected values, we can show the distribution for this one item is:

$$f\left(Y_{pi}|F_p\right) \sim N\left(\mu_i + \lambda_i F_p, \psi_i^2\right)$$

- Therefore, for all I items, our conditional distribution is:

$$f\left(\mathbf{Y}\middle|F_p\right) \sim N_I\left(\boldsymbol{\mu} + \boldsymbol{\Lambda} F_p, \boldsymbol{\Psi}\right)$$

- With multiple factors, this becomes:

$$f(\mathbf{Y}|\mathbf{F}) \sim N_I(\boldsymbol{\mu} + \boldsymbol{\Lambda}\mathbf{F}, \boldsymbol{\Psi})$$

Putting Together the Pieces of Empirical Bayes Factor Scores

$$f(A|B) = \frac{f(B|A)f(A)}{f(B)} = f(\mathbf{F}|\mathbf{Y}) = \frac{f(\mathbf{Y}|\mathbf{F})f(\mathbf{F})}{f(\mathbf{Y})}$$

- For $f(\mathbf{F})$, consider the distribution assumed by the factor:
  - For one factor

$$f\left(F_p\right) \sim N(\mu_F, \sigma_F^2)$$

  - For multiple factors K

$$f(\mathbf{F}) \sim N_K(\boldsymbol{\mu}_F, \boldsymbol{\Phi})$$

- We must pick an identification method which determines if certain parameters of $\boldsymbol{\mu}_F$ and $\boldsymbol{\Phi}$ are fixed or are estimated
  - Any method identification works, so we keep $\boldsymbol{\mu}_F$ and $\boldsymbol{\Phi}$ throughout

Putting Together the Pieces of Empirical Bayes Factor Scores

$$f(A|B) = \frac{f(B|A)f(A)}{f(B)} = f(\mathbf{F}|\mathbf{Y}) = \frac{f(\mathbf{Y}|\mathbf{F})f(\mathbf{F})}{f(\mathbf{Y})}$$

- For $f(\mathbf{Y})$, we return to the model-implied mean vector and covariance matrix:

$$f(\mathbf{Y}) \sim N_I(\boldsymbol{\mu} + \boldsymbol{\Lambda}^T \boldsymbol{\mu}_F, \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi})$$

A Quick Reminder About Types of Distributions

- For two random variables x and z, a conditional distribution is written as: $f(z|x)$
- The conditional distribution is also equal to the joint distribution divided by the marginal distribution of the conditioning random variable
$$f(z|x) = \frac{f(z,x)}{f(x)}$$
- Therefore, the joint distribution can be found by the product of the conditional and marginal distributions:
$$f(z,x) = f(z|x)f(x)$$
- We can use this result in our analysis:
$$f(\mathbf{Y}|\mathbf{F})f(\mathbf{F}) = f(\mathbf{Y},\mathbf{F})$$

A Note about Multivariate Normal Distributions

- If $\mathbf{X}$ is distributed multivariate normally:

Conditional distributions of $\mathbf{X}$ are multivariate normal

- We can show that $f(\mathbf{Y}, \mathbf{F})$, the joint distribution of the data and the factors, is multivariate normal

- We can then use the result above (shown on the next slides) to show that our posterior distribution of the factor scores is also multivariate normal
  - This result **only** applies for measurement models assuming normally distributed data and normally distributed factors: CFA
  - For IRT (and other measurement models), this result will not hold—but this distribution is asymptotically normal as the number of items gets large

Conditional Distributions of MVN Variables are Multivariate Normal

- The conditional distribution of a set of MVN variables is also MVN

- If we were interested in the distribution of the first *q* variables, we partition three matrices:
  - The data: $\left[\mathbf{X}_{1:(N\ x\ q)} \;\middle|\; \mathbf{X}_{2:(N\ x\ p-q)}\right]$
  - The mean vector: $\left[\begin{array}{c} \boldsymbol{\mu}_{1:(q\ x\ 1)} \\ \hline \boldsymbol{\mu}_{2:(p-q\ x\ 1)} \end{array}\right]$
  - The covariance matrix: $\left[\begin{array}{c|c} \boldsymbol{\Sigma}_{11:(q\ x\ q)} & \boldsymbol{\Sigma}_{12:(q\ x\ p-q)} \\ \hline \boldsymbol{\Sigma}_{21:(p-q\ x\ q)} & \boldsymbol{\Sigma}_{22:(p-q\ x\ p-q)} \end{array}\right]$

Conditional Distributions of MVN Variables

- The, $f(\mathbf{X}_1|\mathbf{X}_2)$, conditional distribution of $\mathbf{X}_1$ given the values of $\mathbf{X}_2 = \mathbf{x}_2$ is then:

$$\mathbf{X}_1|\mathbf{X}_2 \sim N_q(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$$

Where (using our partitioned matrices):

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2^T - \boldsymbol{\mu}_\mathbf{2})$$

And:

$$\boldsymbol{\Sigma}^* = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$$

Derive, Templin, Derive!

- The joint distribution of all I items and K factor scores is

$$f(\mathbf{Y},\mathbf{F}) = f\left(\left[\begin{array}{c}\mathbf{Y} \\ \hdashline \mathbf{F}\end{array}\right]\right) = N_{I+K}\left(\left[\begin{array}{c}\boldsymbol{\mu} + \boldsymbol{\Lambda}^T\boldsymbol{\mu}_F \\ \hdashline \boldsymbol{\mu}_F\end{array}\right], \left[\begin{array}{c:c}\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi} & \boldsymbol{\Lambda}\boldsymbol{\Phi} \\ \hdashline \boldsymbol{\Phi}\boldsymbol{\Lambda}^T & \boldsymbol{\Phi}\end{array}\right]\right)$$

- Using the conditional distributions of MVNs result:

$f\left(\mathbf{F}_p \middle| \mathbf{Y}_p\right)$ is MVN:

With mean: $\boldsymbol{\mu}_F + \boldsymbol{\Phi}\boldsymbol{\Lambda}^T\left(\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1}\left(\mathbf{Y}_p^T - \boldsymbol{\mu}\right)$

And Covariance: $\boldsymbol{\Phi} - \boldsymbol{\Phi}\boldsymbol{\Lambda}^T\left(\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{\Phi}$

#WTFTemplin

Implications for Factor Scores

- **When using measurement models assuming normally distributed data and normally distributed factors (CFA)**:
  - The posterior distribution of the factor scores is MVN
  - Therefore, the **most likely** factor score (MAP) and the **expected** factor score (EAP) is given by the mean from the previous slides
  - The factor score is a function of the model parameter estimates and the data

Linking Sum Scores and CTT to Measurement Models Via Factor Scores

Connecting Sum Scores and Factor Scores

- Sum scores have a correlation of 1.0 with factor scores from a **parallel items** CFA model
  - Parallel items model: all factor loadings equal + all unique variances equal

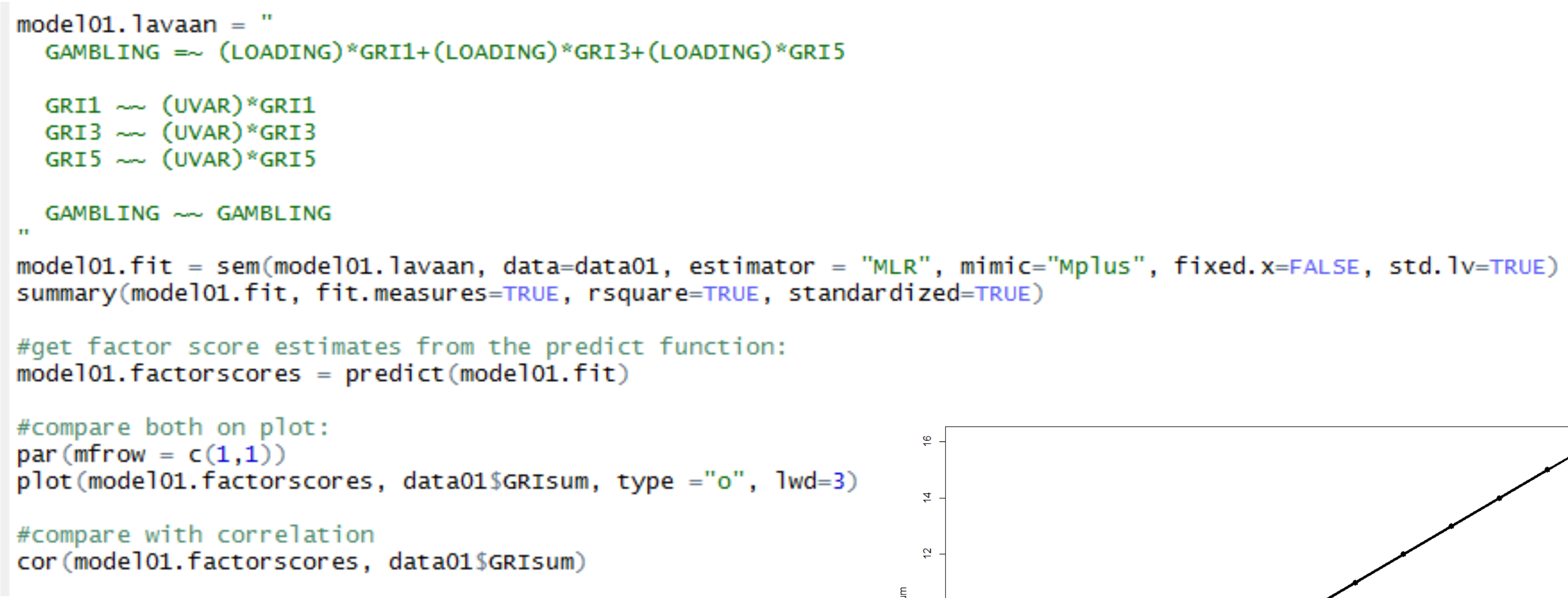
- For example, here are the parallel items model equations for our three-item GRI example data:

$$GRI1_p = \mu_1 + \lambda F_p + e_{p1}; \qquad e_{p1} \sim N(0, \psi^2)$$
$$GRI3_p = \mu_1 + \lambda F_p + e_{p3}; \qquad e_{p3} \sim N(0, \psi^2)$$
$$GRI5_p = \mu_1 + \lambda F_p + e_{p5}; \qquad e_{p5} \sim N(0, \psi^2)$$

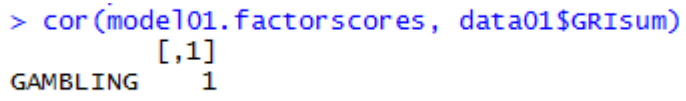
- With a common loading estimated, we will use a standardized factor identification (but we don't have to)

$$F_p \sim N(0, 1)$$

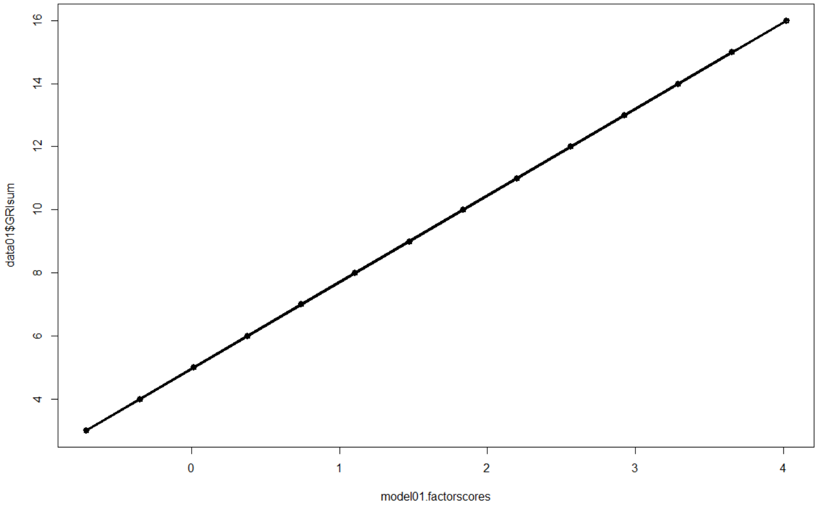
Comparing a Parallel Items Model Factor Score to a Sum Score



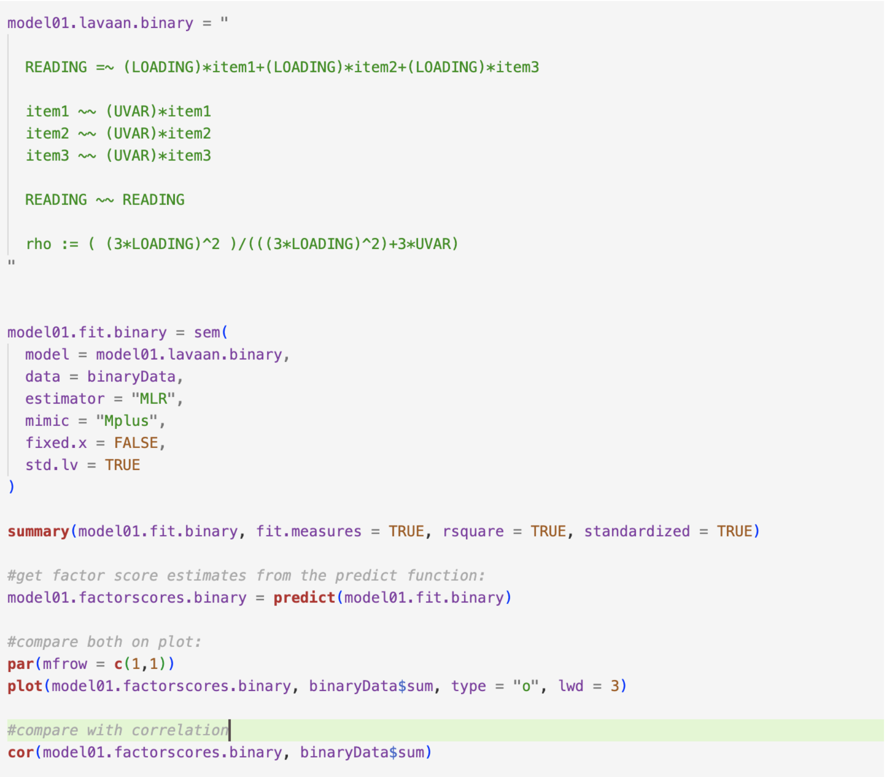
```
model01.lavaan = "
  GAMBLING =~ (LOADING)*GRI1+(LOADING)*GRI3+(LOADING)*GRI5

  GRI1 ~~ (UVAR)*GRI1
  GRI3 ~~ (UVAR)*GRI3
  GRI5 ~~ (UVAR)*GRI5

  GAMBLING ~~ GAMBLING
"
model01.fit = sem(model01.lavaan, data=data01, estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)
summary(model01.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

#get factor score estimates from the predict function:
model01.factorscores = predict(model01.fit)

#compare both on plot:
par(mfrow = c(1,1))
plot(model01.factorscores, data01$GRIsum, type ="o", lwd=3)

#compare with correlation
cor(model01.factorscores, data01$GRIsum)
```



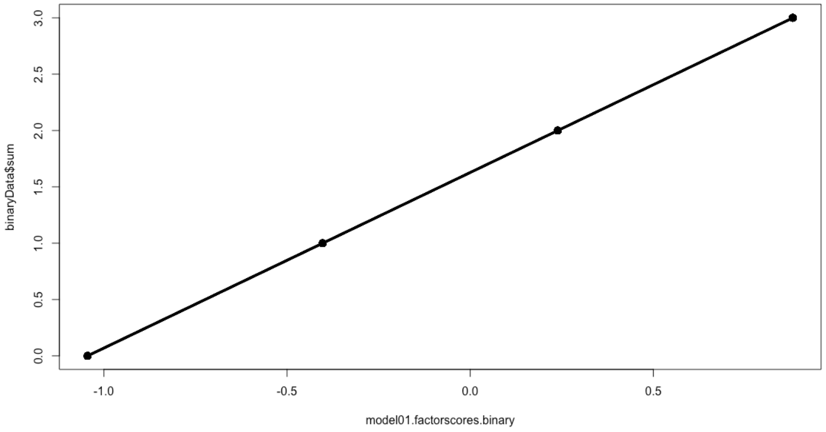
```
> cor(model01.factorscores, data01$GRIsum)
          [,1]
GAMBLING     1
```

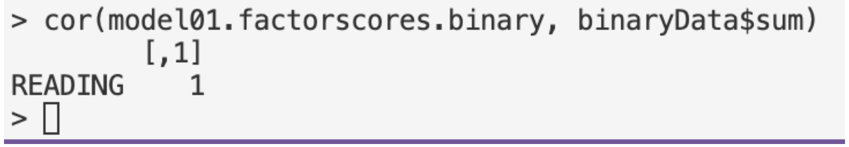


Comparing a Parallel Items Model Factor Score to a Sum Score (Binary Items)



```
model01.lavaan.binary = "

  READING =~ (LOADING)*item1+(LOADING)*item2+(LOADING)*item3

  item1 ~~ (UVAR)*item1
  item2 ~~ (UVAR)*item2
  item3 ~~ (UVAR)*item3

  READING ~~ READING

  rho := ( (3*LOADING)^2 )/(((3*LOADING)^2)+3*UVAR)
"


model01.fit.binary = sem(
  model = model01.lavaan.binary,
  data = binaryData,
  estimator = "MLR",
  mimic = "Mplus",
  fixed.x = FALSE,
  std.lv = TRUE
)

summary(model01.fit.binary, fit.measures = TRUE, rsquare = TRUE, standardized = TRUE)

#get factor score estimates from the predict function:
model01.factorscores.binary = predict(model01.fit.binary)

#compare both on plot:
par(mfrow = c(1,1))
plot(model01.factorscores.binary, binaryData$sum, type = "o", lwd = 3)

#compare with correlation
cor(model01.factorscores.binary, binaryData$sum)
```

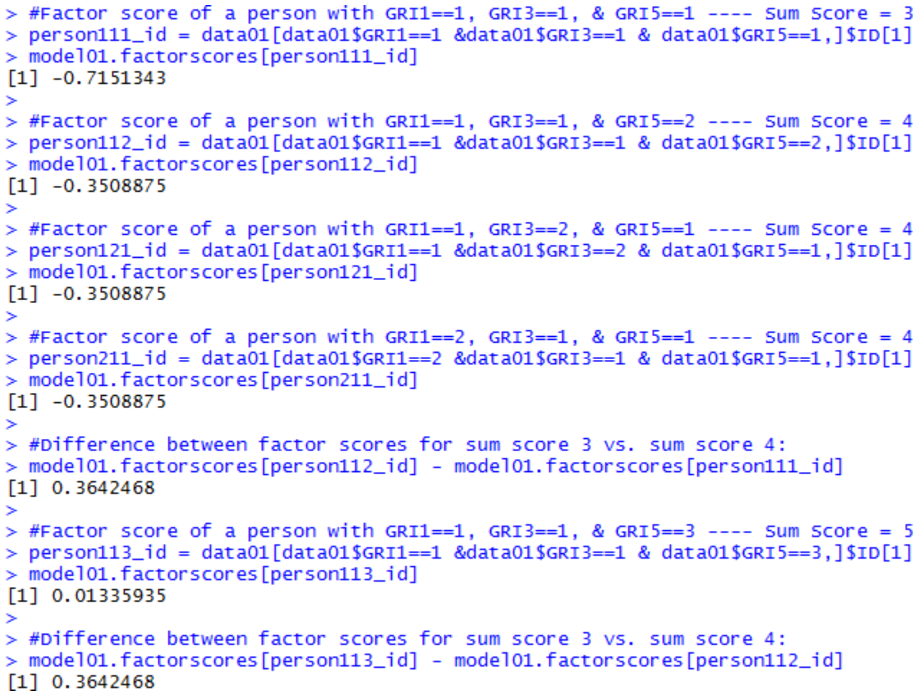




```
> cor(model01.factorscores.binary, binaryData$sum)
        [,1]
READING    1
> 
```

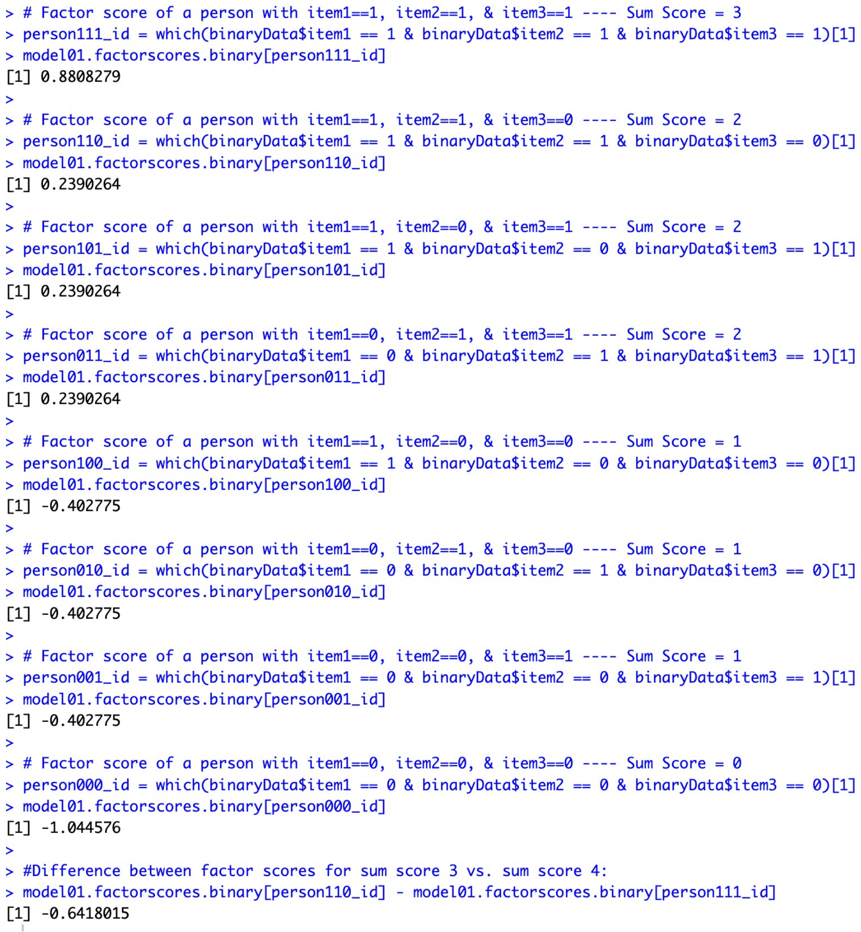
Comparing for Specific Scores

- To look more closely at factor scores versus sum scores, consider the following five people in the Likert item data set



```
> #Factor score of a person with GRI1==1, GRI3==1, & GRI5==1 ---- Sum Score = 3
> person111_id = data01[data01$GRI1==1 &data01$GRI3==1 & data01$GRI5==1,]$ID[1]
> model01.factorscores[person111_id]
[1] -0.7151343
>
> #Factor score of a person with GRI1==1, GRI3==1, & GRI5==2 ---- Sum Score = 4
> person112_id = data01[data01$GRI1==1 &data01$GRI3==1 & data01$GRI5==2,]$ID[1]
> model01.factorscores[person112_id]
[1] -0.3508875
>
> #Factor score of a person with GRI1==1, GRI3==2, & GRI5==1 ---- Sum Score = 4
> person121_id = data01[data01$GRI1==1 &data01$GRI3==2 & data01$GRI5==1,]$ID[1]
> model01.factorscores[person121_id]
[1] -0.3508875
>
> #Factor score of a person with GRI1==2, GRI3==1, & GRI5==1 ---- Sum Score = 4
> person211_id = data01[data01$GRI1==2 &data01$GRI3==1 & data01$GRI5==1,]$ID[1]
> model01.factorscores[person211_id]
[1] -0.3508875
>
> #Difference between factor scores for sum score 3 vs. sum score 4:
> model01.factorscores[person112_id] - model01.factorscores[person111_id]
[1] 0.3642468
>
> #Factor score of a person with GRI1==1, GRI3==1, & GRI5==3 ---- Sum Score = 5
> person113_id = data01[data01$GRI1==1 &data01$GRI3==1 & data01$GRI5==3,]$ID[1]
> model01.factorscores[person113_id]
[1] 0.01335935
>
> #Difference between factor scores for sum score 3 vs. sum score 4:
> model01.factorscores[person113_id] - model01.factorscores[person112_id]
[1] 0.3642468
```

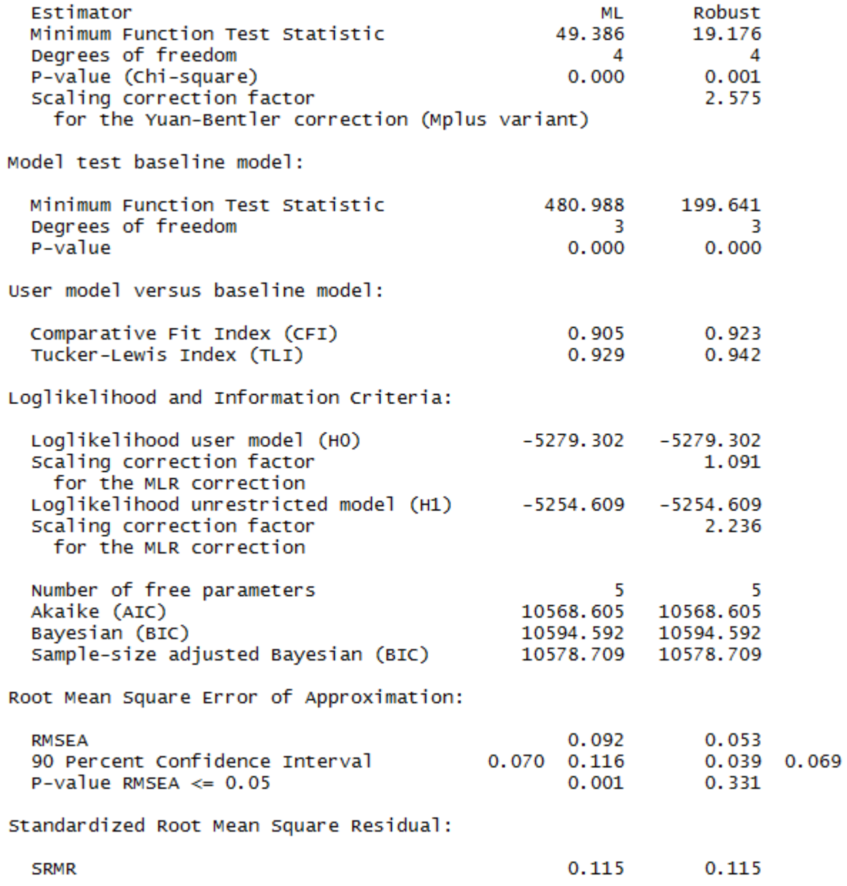
Comparing Scores: Binary Data



```
> # Factor score of a person with item1==1, item2==1, & item3==1 ---- Sum Score = 3
> person111_id = which(binaryData$item1 == 1 & binaryData$item2 == 1 & binaryData$item3 == 1)[1]
> model01.factorscores.binary[person111_id]
[1] 0.8808279
>
> # Factor score of a person with item1==1, item2==1, & item3==0 ---- Sum Score = 2
> person110_id = which(binaryData$item1 == 1 & binaryData$item2 == 1 & binaryData$item3 == 0)[1]
> model01.factorscores.binary[person110_id]
[1] 0.2390264
>
> # Factor score of a person with item1==1, item2==0, & item3==1 ---- Sum Score = 2
> person101_id = which(binaryData$item1 == 1 & binaryData$item2 == 0 & binaryData$item3 == 1)[1]
> model01.factorscores.binary[person101_id]
[1] 0.2390264
>
> # Factor score of a person with item1==0, item2==1, & item3==1 ---- Sum Score = 2
> person011_id = which(binaryData$item1 == 0 & binaryData$item2 == 1 & binaryData$item3 == 1)[1]
> model01.factorscores.binary[person011_id]
[1] 0.2390264
>
> # Factor score of a person with item1==1, item2==0, & item3==0 ---- Sum Score = 1
> person100_id = which(binaryData$item1 == 1 & binaryData$item2 == 0 & binaryData$item3 == 0)[1]
> model01.factorscores.binary[person100_id]
[1] -0.402775
>
> # Factor score of a person with item1==0, item2==1, & item3==0 ---- Sum Score = 1
> person010_id = which(binaryData$item1 == 0 & binaryData$item2 == 1 & binaryData$item3 == 0)[1]
> model01.factorscores.binary[person010_id]
[1] -0.402775
>
> # Factor score of a person with item1==0, item2==0, & item3==1 ---- Sum Score = 1
> person001_id = which(binaryData$item1 == 0 & binaryData$item2 == 0 & binaryData$item3 == 1)[1]
> model01.factorscores.binary[person001_id]
[1] -0.402775
>
> # Factor score of a person with item1==0, item2==0, & item3==0 ---- Sum Score = 0
> person000_id = which(binaryData$item1 == 0 & binaryData$item2 == 0 & binaryData$item3 == 0)[1]
> model01.factorscores.binary[person000_id]
[1] -1.044576
>
> #Difference between factor scores for sum score 3 vs. sum score 4:
> model01.factorscores.binary[person110_id] - model01.factorscores.binary[person111_id]
[1] -0.6418015
```

Before We Get Too Far…
Did The Model Fit?

- Good model fit…
- We could use the model
- So, we could use the factor scores or the sum scores
- But we won't!



```
  Estimator                                         ML      Robust
  Minimum Function Test Statistic               49.386      19.176
  Degrees of freedom                                 4           4
  P-value (Chi-square)                           0.000       0.001
  Scaling correction factor                                  2.575
    for the Yuan-Bentler correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic              480.988     199.641
  Degrees of freedom                                 3           3
  P-value                                        0.000       0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                    0.905       0.923
  Tucker-Lewis Index (TLI)                       0.929       0.942

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -5279.302   -5279.302
  Scaling correction factor                                  1.091
    for the MLR correction
  Loglikelihood unrestricted model (H1)      -5254.609   -5254.609
  Scaling correction factor                                  2.236
    for the MLR correction

  Number of free parameters                          5           5
  Akaike (AIC)                               10568.605   10568.605
  Bayesian (BIC)                             10594.592   10594.592
  Sample-size adjusted Bayesian (BIC)        10578.709   10578.709

Root Mean Square Error of Approximation:

  RMSEA                                          0.092       0.053
  90 Percent Confidence Interval          0.070  0.116       0.039  0.069
  P-value RMSEA <= 0.05                          0.001       0.331

Standardized Root Mean Square Residual:

  SRMR                                           0.115       0.115
```

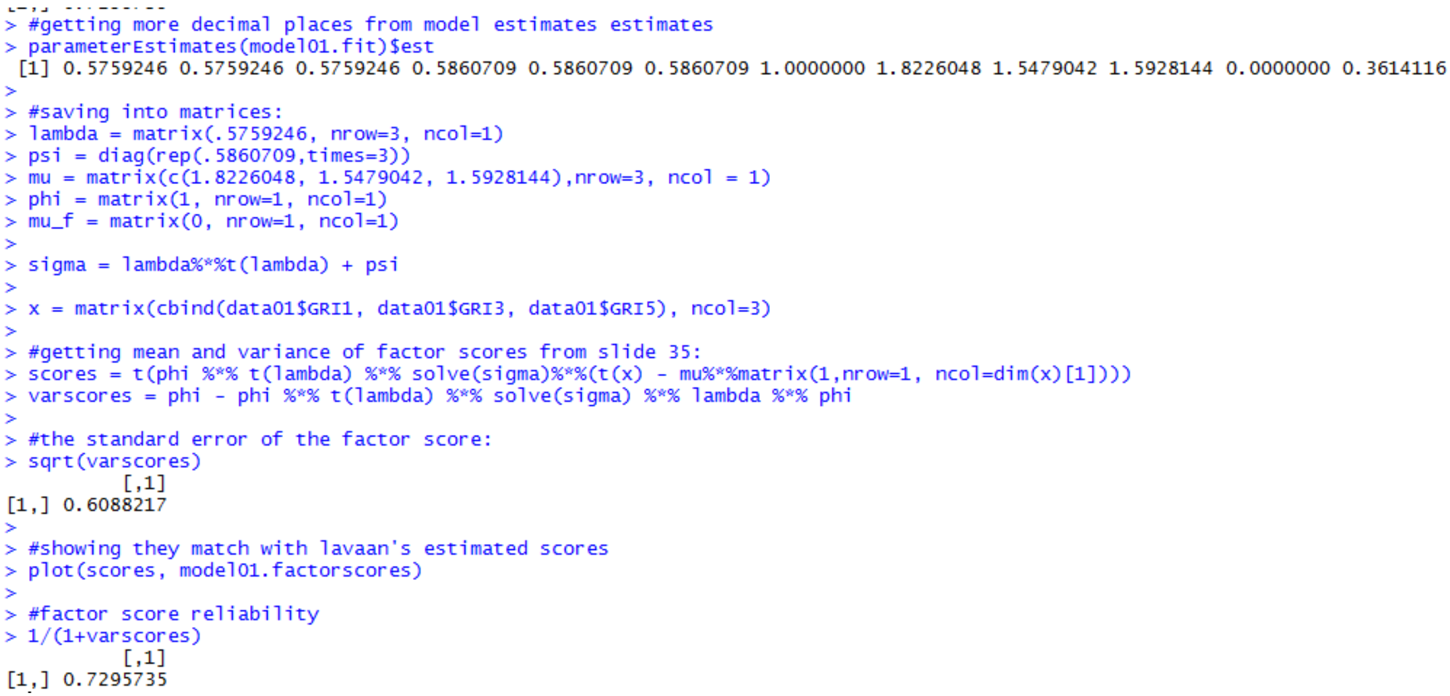
And…About Reliability

- Factor score reliability is:

$$\rho = \frac{\sigma_F^2}{\sigma_F^2 + SE\left(F_p\right)^2}$$

- lavaan does not compute the factor score standard errors (Mplus does)…but that's okay, because we can grab them from the matrix algebra on p. 35

R Syntax for Computing SE of Factor Scores



```
> #getting more decimal places from model estimates estimates
> parameterEstimates(model01.fit)$est
 [1] 0.5759246 0.5759246 0.5759246 0.5860709 0.5860709 0.5860709 1.0000000 1.8226048 1.5479042 1.5928144 0.0000000 0.3614116
>
> #saving into matrices:
> lambda = matrix(.5759246, nrow=3, ncol=1)
> psi = diag(rep(.5860709,times=3))
> mu = matrix(c(1.8226048, 1.5479042, 1.5928144),nrow=3, ncol = 1)
> phi = matrix(1, nrow=1, ncol=1)
> mu_f = matrix(0, nrow=1, ncol=1)
>
> sigma = lambda%*%t(lambda) + psi
>
> x = matrix(cbind(data01$GRI1, data01$GRI3, data01$GRI5), ncol=3)
>
> #getting mean and variance of factor scores from slide 35:
> scores = t(phi %*% t(lambda) %*% solve(sigma)%*%(t(x) - mu%*%matrix(1,nrow=1, ncol=dim(x)[1])))
> varscores = phi - phi %*% t(lambda) %*% solve(sigma) %*% lambda %*% phi
>
> #the standard error of the factor score:
> sqrt(varscores)
          [,1]
[1,] 0.6088217
>
> #showing they match with lavaan's estimated scores
> plot(scores, model01.factorscores)
>
> #factor score reliability
> 1/(1+varscores)
          [,1]
[1,] 0.7295735
```

Reliability of Factor Score = .73

What about the reliability of our sum scores?

Classical Test Theory from a CFA Perspective

- In CTT the unit of analysis is the test score:

$$Y_{p,Total} = T_p + E_p$$

- In CFA the unit of analysis is the item:

$$Y_{pi} = \mu_{I_i} + \lambda_i F_p + e_{pi}$$

- To map CFA onto CTT, we must put these together:

$$Y_{p,Total} = \sum_{i=1}^{I} Y_{pi}$$

Further Unpacking of the Total Score Formula

- Because CFA is an item-based model, we can then substitute each item's model into the sum:

$$Y_{p,Total} = \sum_{i=1}^{I} Y_{pi} = \sum_{i=1}^{I} \left(\mu_{I_i} + \lambda_i F_p + e_{pi}\right)$$

$$= \sum_{i=1}^{I} \mu_{I_i} + \left(\sum_{i=1}^{I} \lambda_i\right) F_p + \sum_{i=1}^{I} e_{pi}$$

- Mapping this onto true score and error from CTT:

$$T = \sum_{i=1}^{I} \mu_{I_i} + \left(\sum_{i=1}^{I} \lambda_i\right) F_p \text{ and } E = \sum_{i=1}^{I} e_{pi}$$

CFA-Model Estimated Reliability of Sum Scores

- From:

$$T = \sum_{i=1}^{I} \mu_{I_i} + \left(\sum_{i=1}^{I} \lambda_i \right) F_p \text{ and } E = \sum_{i=1}^{I} e_{pi}$$

- $Var(T) = Var\left(\sum_{i=1}^{I} \mu_{I_i}\right) + \left(\sum_{i=1}^{I} \lambda_i^2\right) Var\left(F_p\right) =$

$$\left(\sum_{i=1}^{I} \lambda_i \right)^2 \sigma_F^2$$

- $Var(E) = Var\left(\sum_{i=1}^{I} e_{pi}\right) =$

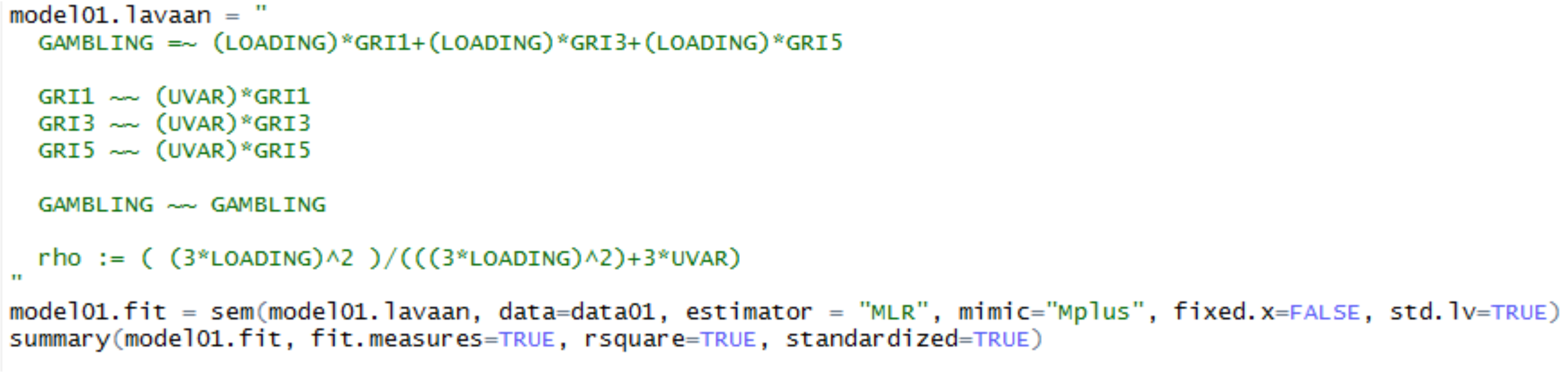
$$\sum_{i=1}^{I} \psi_i^2$$

For models with correlated residuals, those add to Var(E)

CFA-Model Estimated Reliability of Sum Scores

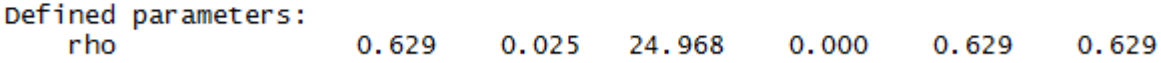
- From the previous slide:

$$\rho = \frac{Var(T)}{Var(T) + Var(E)} = \frac{\left(\sum_{i=1}^{I} \lambda_i\right)^2 \sigma_F^2}{\left(\sum_{i=1}^{I} \lambda_i\right)^2 \sigma_F^2 + \sum_{i=1}^{I} \psi_i^2}$$



```
model01.lavaan = "
  GAMBLING =~ (LOADING)*GRI1+(LOADING)*GRI3+(LOADING)*GRI5

  GRI1 ~~ (UVAR)*GRI1
  GRI3 ~~ (UVAR)*GRI3
  GRI5 ~~ (UVAR)*GRI5

  GAMBLING ~~ GAMBLING

  rho := ( (3*LOADING)^2 )/(((3*LOADING)^2)+3*UVAR)
"
model01.fit = sem(model01.lavaan, data=data01, estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)
summary(model01.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

- The estimated reliability is….



```
Defined parameters:
   rho                0.629    0.025   24.968    0.000    0.629    0.629
```

Notes on CFA-Estimated Reliabilities

- The CFA-Estimated reliability is for the sum score, not the factor score

- The sum score's reliability is .629 (SE = .025); the factor score's reliability is .73
  - The difference comes from additional sources of error in the factor score:
    - Sampling error
    - Error from the prior distribution (squishing the variance of the factor/error)

- The sum score's reliability is equal to the **Spearman Brown** reliability estimate
  - Therefore, CTT reliability estimates can come from CFA….

Comparing Other CFA Models with Sum Scores

- Another model to consider is the Tau-equivalent items model, which, for CFA, means equal loadings but different unique variances:

- For example, here are the parallel items model equations for our three-item GRI example data:

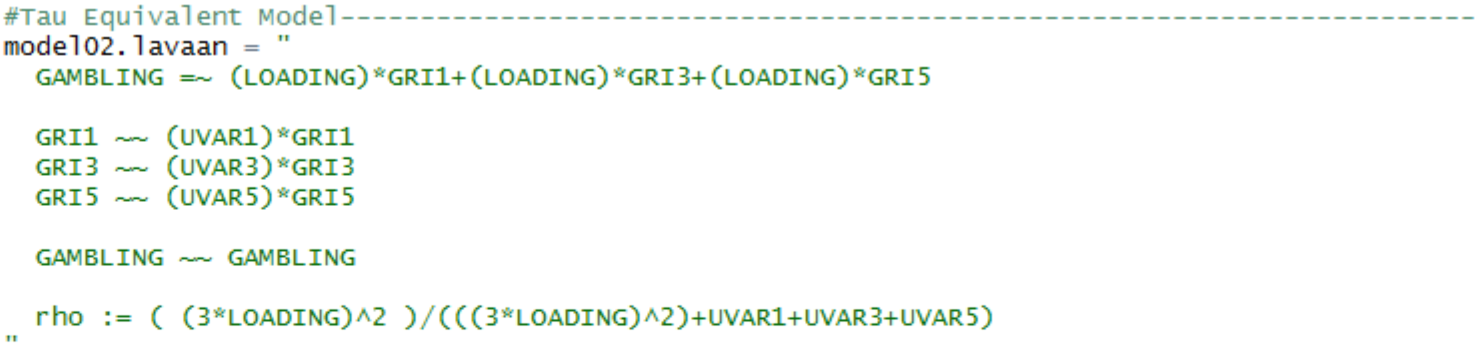
$$GRI1_p = \mu_1 + \lambda F_p + e_{p1}; \qquad e_{p1} \sim N(0, \psi_1^2)$$
$$GRI3_p = \mu_1 + \lambda F_p + e_{p3}; \qquad e_{p3} \sim N(0, \psi_3^2)$$
$$GRI5_p = \mu_1 + \lambda F_p + e_{p5}; \qquad e_{p5} \sim N\left(0, \psi_5^2\right)$$

- With a common loading estimated, we will use a standardized factor identification (but we don't have to)

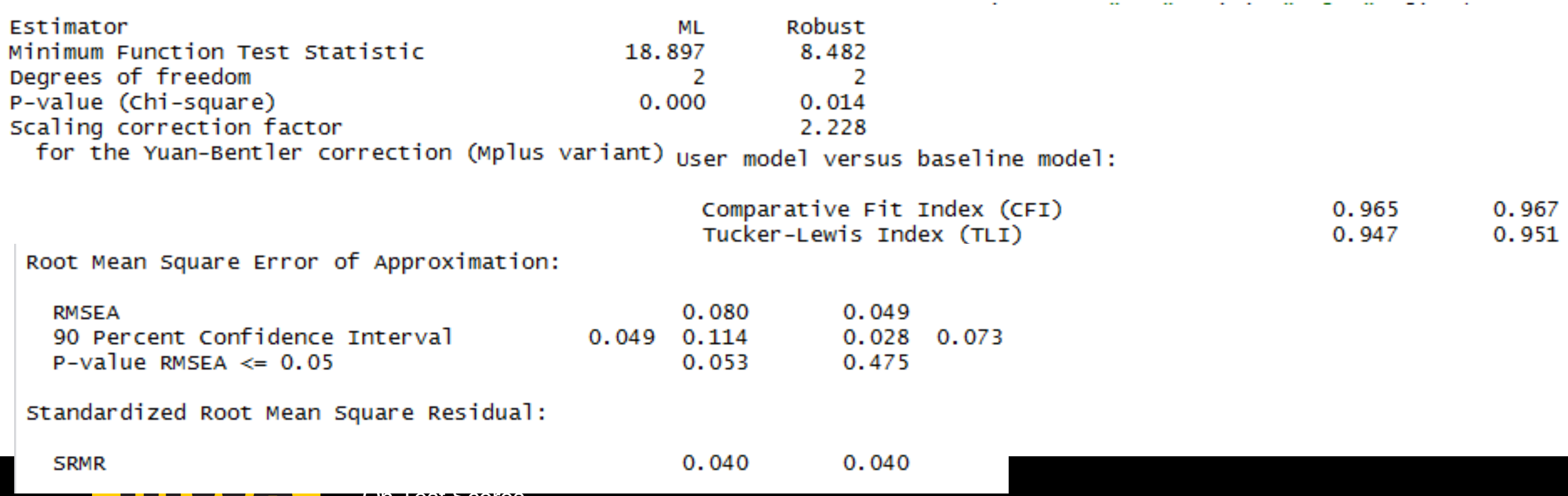
$$F_p \sim N(0,1)$$

The Tau Equivalent Model in lavaan

- Note: shown for didactic purposes (don’t use this model)



```
#Tau Equivalent Model-------------------------------------------------------------------
model02.lavaan = "
  GAMBLING =~ (LOADING)*GRI1+(LOADING)*GRI3+(LOADING)*GRI5

  GRI1 ~~ (UVAR1)*GRI1
  GRI3 ~~ (UVAR3)*GRI3
  GRI5 ~~ (UVAR5)*GRI5

  GAMBLING ~~ GAMBLING

  rho := ( (3*LOADING)^2 )/(((3*LOADING)^2)+UVAR1+UVAR3+UVAR5)
"
```

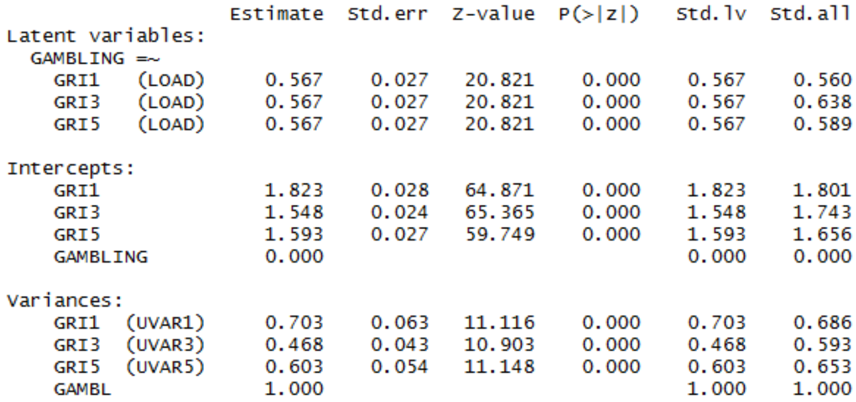


```
Estimator                                         ML       Robust
Minimum Function Test Statistic               18.897        8.482
Degrees of freedom                                 2            2
P-value (Chi-square)                           0.000        0.014
Scaling correction factor                                   2.228
  for the Yuan-Bentler correction (Mplus variant)
```

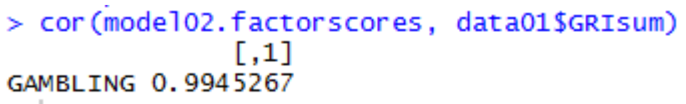
```
User model versus baseline model:

  Comparative Fit Index (CFI)                    0.965       0.967
  Tucker-Lewis Index (TLI)                       0.947       0.951
```

```
Root Mean Square Error of Approximation:

  RMSEA                                          0.080       0.049
  90 Percent Confidence Interval          0.049  0.114       0.028  0.073
  P-value RMSEA <= 0.05                          0.053       0.475

Standardized Root Mean Square Residual:

  SRMR                                           0.040       0.040
```

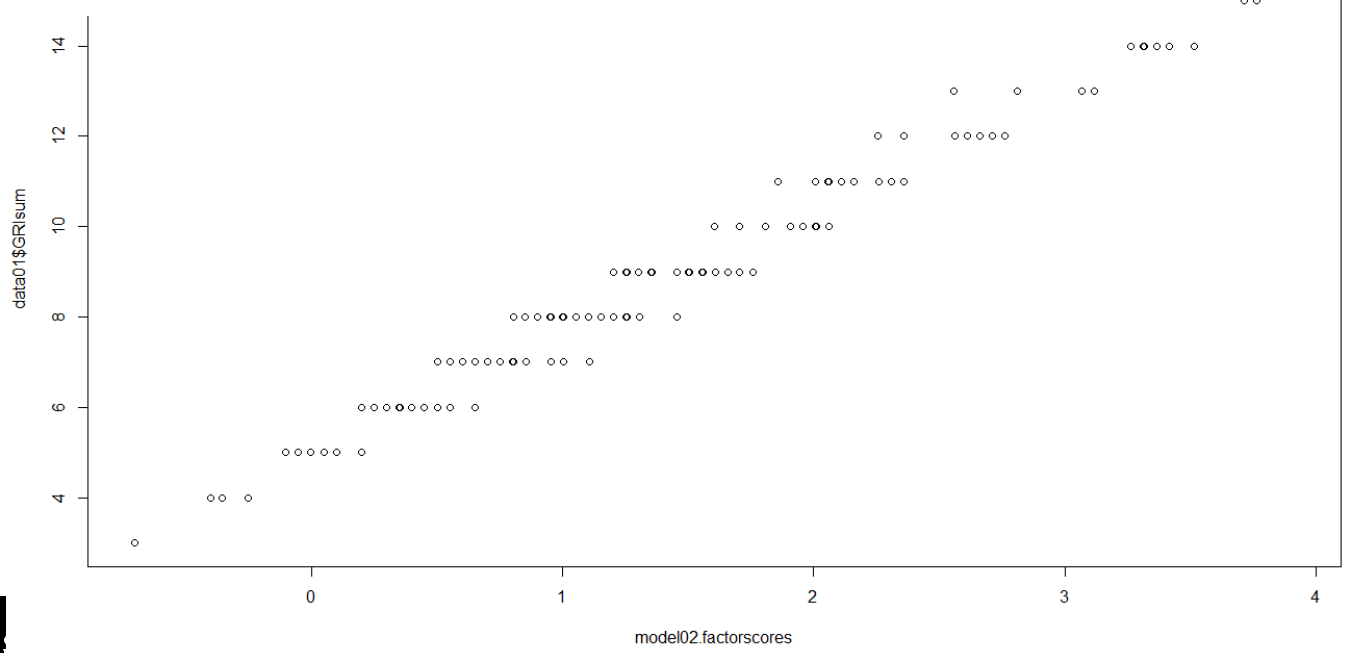
Parameter Estimates vs. Factor Score vs. Sum Score



```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  GAMBLING =~
    GRI1    (LOAD)    0.567    0.027   20.821    0.000    0.567    0.560
    GRI3    (LOAD)    0.567    0.027   20.821    0.000    0.567    0.638
    GRI5    (LOAD)    0.567    0.027   20.821    0.000    0.567    0.589

Intercepts:
    GRI1              1.823    0.028   64.871    0.000    1.823    1.801
    GRI3              1.548    0.024   65.365    0.000    1.548    1.743
    GRI5              1.593    0.027   59.749    0.000    1.593    1.656
    GAMBLING          0.000                               0.000    0.000

Variances:
    GRI1   (UVAR1)    0.703    0.063   11.116    0.000    0.703    0.686
    GRI3   (UVAR3)    0.468    0.043   10.903    0.000    0.468    0.593
    GRI5   (UVAR5)    0.603    0.054   11.148    0.000    0.603    0.653
    GAMBL             1.000                               1.000    1.000
```

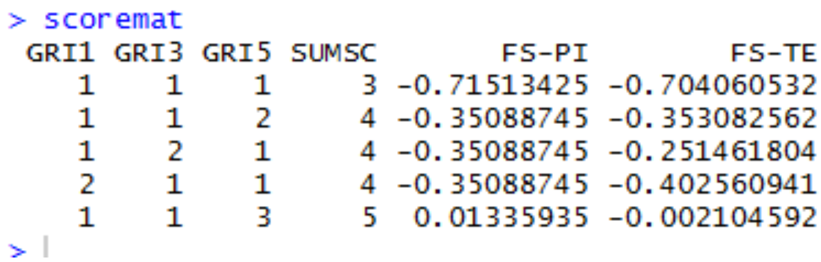


```
> cor(model02.factorscores, data01$GRIsum)
                [,1]
GAMBLING 0.9945267
```

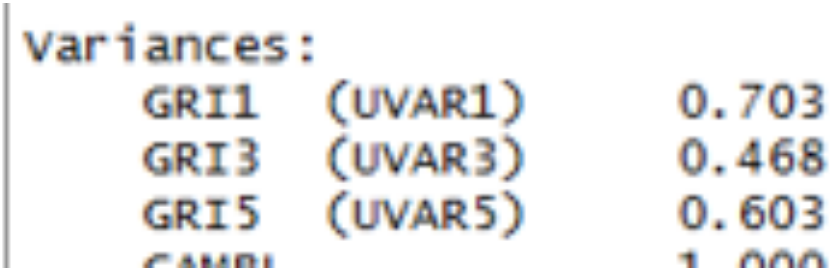


Factor vs. Sum Score…by item

- Now what matters is which item had a higher score…
  - Items with higher information (loading^2/unique variance) result in bigger jumps in factor score relative to items with lower information



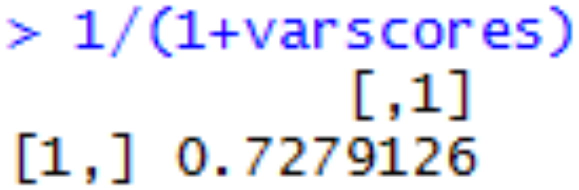
```
> scoremat
 GRI1 GRI3 GRI5 SUMSC       FS-PI        FS-TE
    1    1    1     3 -0.71513425 -0.704060532
    1    1    2     4 -0.35088745 -0.353082562
    1    2    1     4 -0.35088745 -0.251461804
    2    1    1     4 -0.35088745 -0.402560941
    1    1    3     5  0.01335935 -0.002104592
> 
```



```
Variances:
    GRI1    (UVAR1)       0.703
    GRI3    (UVAR3)       0.468
    GRI5    (UVAR5)       0.603
```

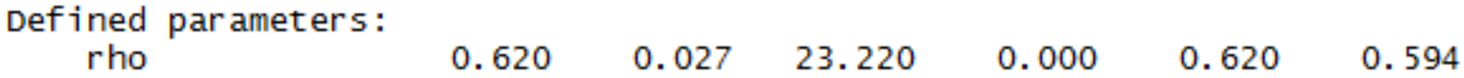
Tau Equivalent Reliability for Factor and Sum Scores

- Factor score reliability estimate: .73



```
> 1/(1+varscores)
          [,1]
[1,] 0.7279126
```

- Sum score reliability estimate: .62



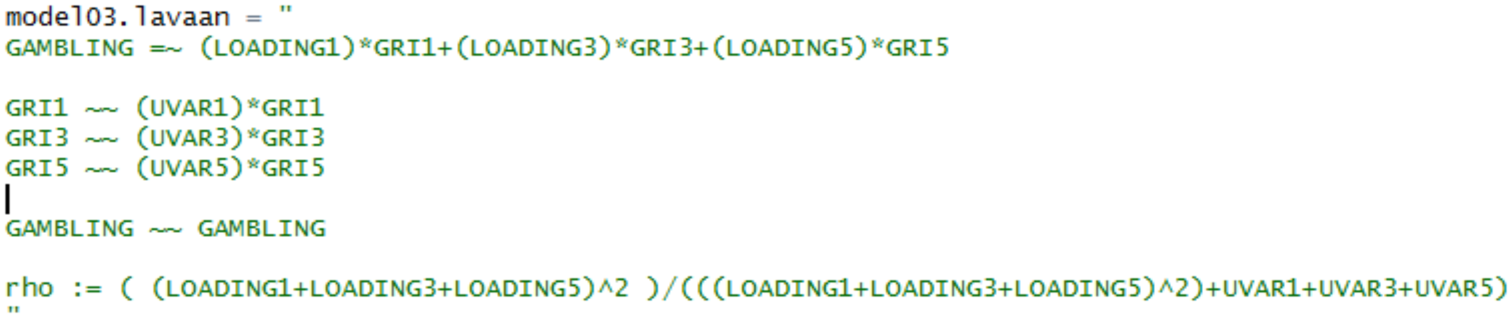
```
Defined parameters:
    rho               0.620    0.027   23.220    0.000    0.620    0.594
```

- The sum score reliability is actually coefficient alpha
  - Cronbach's alpha (1951) / Guttman's Lambda 6 (1945)

- HUGE NOTE: THIS IS WHY RELIABILTY IS NOT AN INDEX OF MODEL FIT
  - IT CAN BE SHOWN TO DEPEND ON PARAMETERS THAT WILL BE BIASED UNDER MISFITTING MODELS

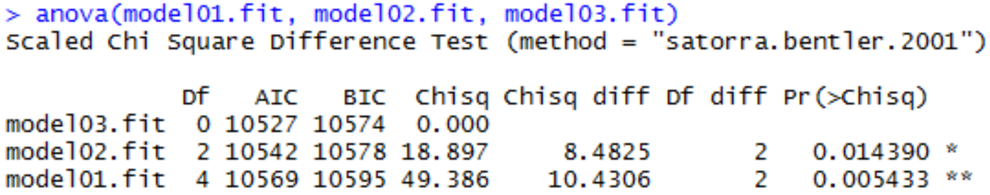
Finally…the Unrestricted CFA Model

- All of the previous slides were to get us to see the relationship between sum scores and CFA models
  - We would never estimate either…we would use an unrestricted CFA model
  - Here is what happens with an that unrestricted CFA model



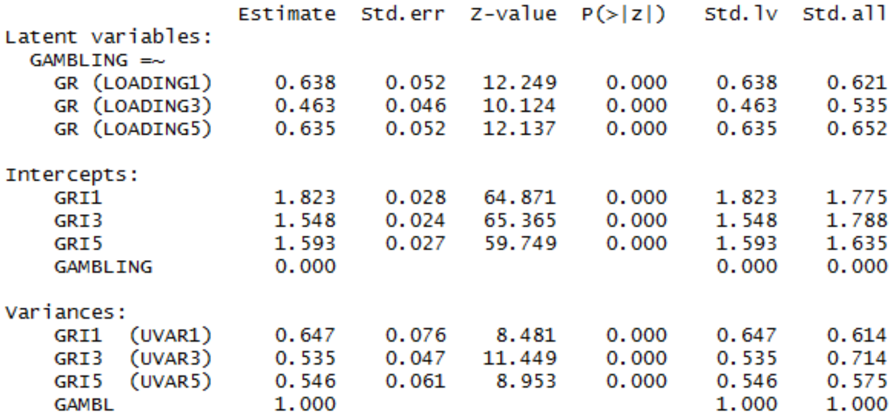
```
model03.lavaan = "
GAMBLING =~ (LOADING1)*GRI1+(LOADING3)*GRI3+(LOADING5)*GRI5

GRI1 ~~ (UVAR1)*GRI1
GRI3 ~~ (UVAR3)*GRI3
GRI5 ~~ (UVAR5)*GRI5

GAMBLING ~~ GAMBLING

rho := ( (LOADING1+LOADING3+LOADING5)^2 )/(((LOADING1+LOADING3+LOADING5)^2)+UVAR1+UVAR3+UVAR5)
"
```

- This model fits perfectly—so no need to check model fit
- Compared to the other two models (we reject CTT)

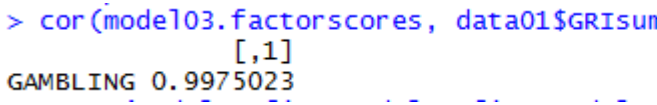


```
> anova(model01.fit, model02.fit, model03.fit)
Scaled Chi Square Difference Test (method = "satorra.bentler.2001")

            Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
model03.fit  0 10527 10574  0.000
model02.fit  2 10542 10578 18.897     8.4825       2   0.014390 *
model01.fit  4 10569 10595 49.386    10.4306       2   0.005433 **
```

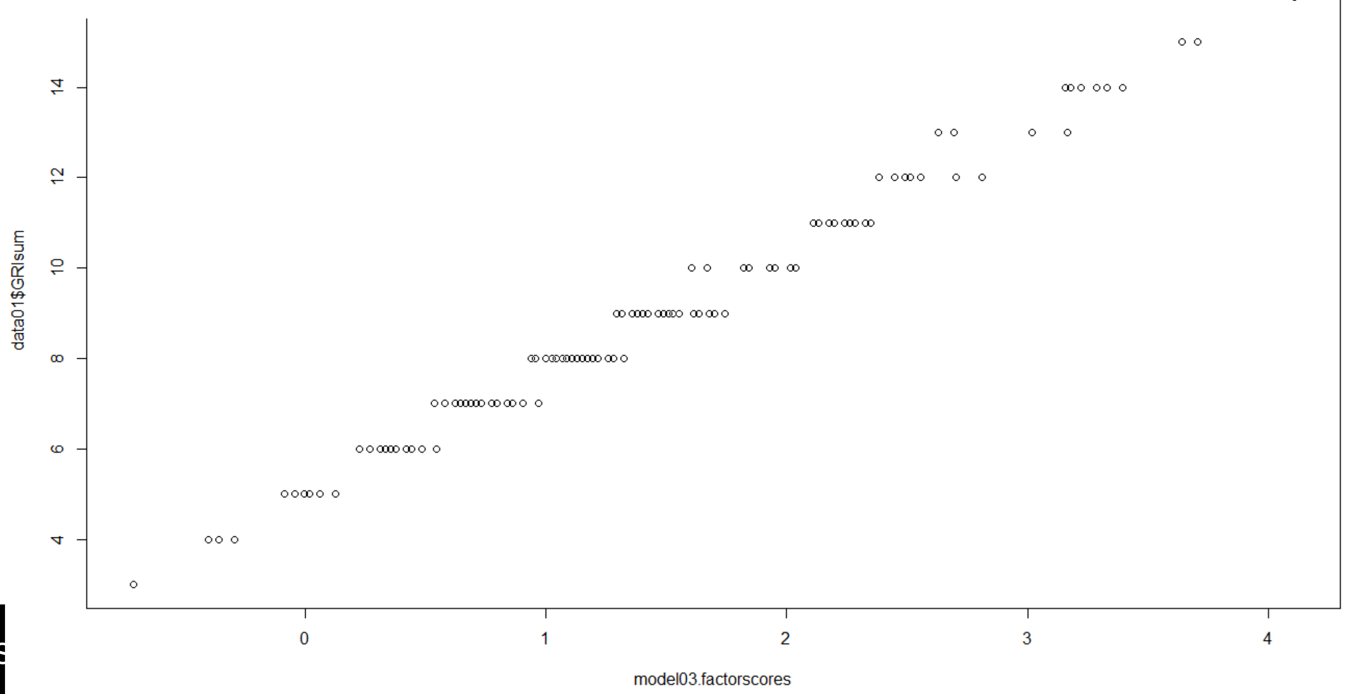
Parameter Estimates vs. Factor Score vs. Sum Score



```
                       Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  GAMBLING =~
    GR (LOADING1)         0.638    0.052   12.249    0.000    0.638    0.621
    GR (LOADING3)         0.463    0.046   10.124    0.000    0.463    0.535
    GR (LOADING5)         0.635    0.052   12.137    0.000    0.635    0.652

Intercepts:
    GRI1                  1.823    0.028   64.871    0.000    1.823    1.775
    GRI3                  1.548    0.024   65.365    0.000    1.548    1.788
    GRI5                  1.593    0.027   59.749    0.000    1.593    1.635
    GAMBLING              0.000                               0.000    0.000

Variances:
    GRI1   (UVAR1)        0.647    0.076    8.481    0.000    0.647    0.614
    GRI3   (UVAR3)        0.535    0.047   11.449    0.000    0.535    0.714
    GRI5   (UVAR5)        0.546    0.061    8.953    0.000    0.546    0.575
    GAMBL                 1.000                               1.000    1.000
```

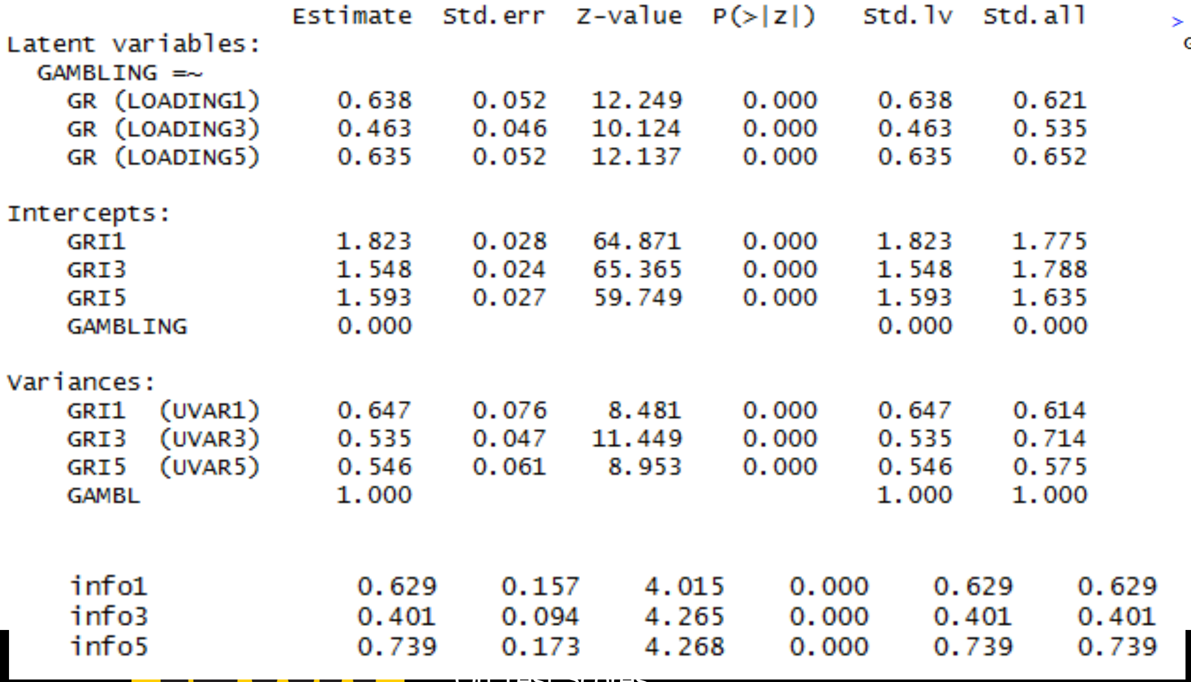


```
> cor(model03.factorscores, data01$GRIsum
              [,1]
GAMBLING 0.9975023
```

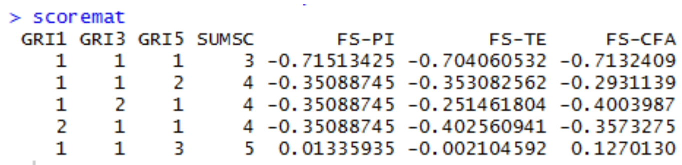


Factor Scores by Sum Score…by item

- Now what matters is which item had a higher score…
  - Items with higher information (loading^2/unique variance) result in bigger jumps in factor score relative to items with lower information



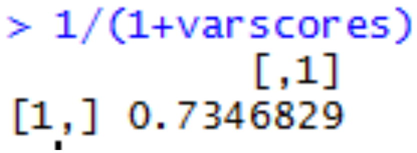
```
                    Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  GAMBLING =~
    GR (LOADING1)      0.638    0.052   12.249    0.000    0.638    0.621
    GR (LOADING3)      0.463    0.046   10.124    0.000    0.463    0.535
    GR (LOADING5)      0.635    0.052   12.137    0.000    0.635    0.652

Intercepts:
    GRI1               1.823    0.028   64.871    0.000    1.823    1.775
    GRI3               1.548    0.024   65.365    0.000    1.548    1.788
    GRI5               1.593    0.027   59.749    0.000    1.593    1.635
    GAMBLING           0.000                               0.000    0.000

Variances:
    GRI1   (UVAR1)     0.647    0.076    8.481    0.000    0.647    0.614
    GRI3   (UVAR3)     0.535    0.047   11.449    0.000    0.535    0.714
    GRI5   (UVAR5)     0.546    0.061    8.953    0.000    0.546    0.575
    GAMBL              1.000                               1.000    1.000


    info1                 0.629    0.157    4.015    0.000    0.629    0.629
    info3                 0.401    0.094    4.265    0.000    0.401    0.401
    info5                 0.739    0.173    4.268    0.000    0.739    0.739
```



```
> scoremat
 GRI1 GRI3 GRI5 SUMSC       FS-PI        FS-TE     FS-CFA
    1    1    1     3 -0.71513425 -0.704060532 -0.7132409
    1    1    2     4 -0.35088745 -0.353082562 -0.2931139
    1    2    1     4 -0.35088745 -0.251461804 -0.4003987
    2    1    1     4 -0.35088745 -0.402560941 -0.3573275
    1    1    3     5  0.01335935 -0.002104592  0.1270130
```

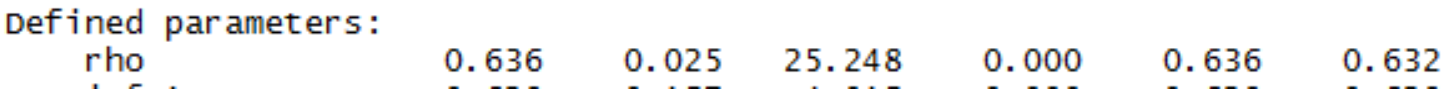
CFA Equivalent Reliability for Factor and Sum Scores

- Factor score reliability estimate: .734



```
> 1/(1+varscores)
          [,1]
[1,] 0.7346829
```

- Sum score reliability estimate: .636



```
Defined parameters:
   rho              0.636    0.025   25.248    0.000    0.636    0.632
```

- The sum score reliability is sometimes called coefficient omega (see McDonald, 1999)

- If all three models fit the data then

Omega > Alpha > Spearman Brown

But…the differences are very small (if the model fits)

Potential Sources of Error in Factor Scores

- Measurement error
  - e.g., the $SE(\hat{F})$

- Model misspecification error of various types:
  - ~~Dimensionality misspecification error~~
    - ~~e.g., Assuming one dimension when there is more than one present~~
  - ~~Parameter constraint misspecification error~~
    - ~~e.g., Assuming overly restrictive constraints (see next section and all of CTT)~~
  - ~~Linear model functional misspecification error~~
    - ~~e.g., Assuming a linear relationship between the factor and the items when a non-linear one is present~~
  - Outcome distribution misspecification error
    - e.g., Assuming Likert-type data to be continuous and using a normal distribution
  - Factor distribution misspecification error
    - e.g., Assuming your trait is normally distributed when it is categorical or a mixture distribution

- ~~Missing data error~~
  - ~~How you treat missing responses to items makes even more untenable assumptions~~

- Sampling error

- Prior Distribution Error
  - e.g., factor scores are “shrunken estimates”

So….?

- Up to this point we have seen
  - Assumptions underlying sum scores
  - Definitions of factor scores
  - How sum scores imply a very specific CFA model

- We have also seen a history of reliability:
  - Spearman Brown (1910): Parallel items model
    - Equal loadings/unique variances
  - Guttman/Cronbch Alpha (1945,1953): Tau equivalent items model
    - Equal loadings
  - Coefficient omega (source unknown): Unrestricted CFA model
  - Reliability for factor scores
  - Also note: the next step is conditional reliability (IRT models)

- **The point is that if you are ever reporting scores but not using them in subsequent analyses, then use a factor score**

- But what we haven't seen is what to do when we cannot use a simultaneous analysis/SEM
  - And that answer will have to come during the next lecture…

Error Score Card

Error Source	Sum Score	Factor Score
Measurement Error	🟥	🟥
Sampling Error	🟩	🟥
Prior Distribution Bias	🟩	🟥
ML Estimator Bias	🟩	🟥
MSPFD: Dimensionality	🟥	🟩
MSPFD: Constraints	🟥	🟩
MSPFD: Lin Pred Function	🟥	🟩
MSPFD: Data Distribution	🟥	🟥
MSPFD: Factor Distribution	🟥	🟥

Red: Source of Error Potentially Present **Green: Not a Source of Error**

Don’t count red-vs-green as some types of error is worse than others

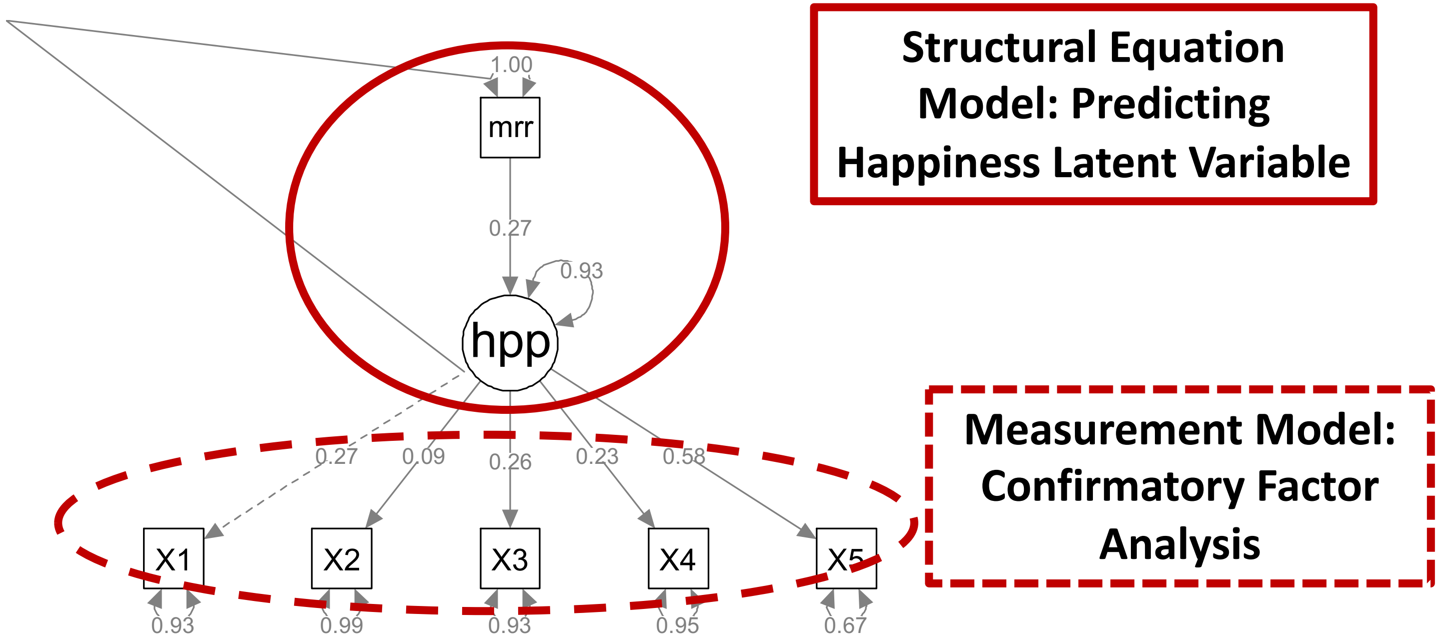
Secondary Analyses with Estimated Scores

Example Data, Part Two…

- To introduce and motivate SEM, and to review some prerequisites, we will make use of an example data set
- Data come from a (simulated) sample of 150 participants who provided self-reports of a happiness scale and their marital status
- Participant responded one survey:
  - 5-item happiness survey (each item used roughly a 5-point Likert scale)
  - 1-item marital status question (are you married? Yes/No)
- The researchers were interested in the effects of marital status on happiness

In SEM, We Don’t Need a Sum Score

- Variables that are measured with error are represented as “latent” variables in SEM
  - The latent variables are estimated directly by the model
  - Any equations involving latent variables are estimated simultaneously
- A more accurate depiction of our example:

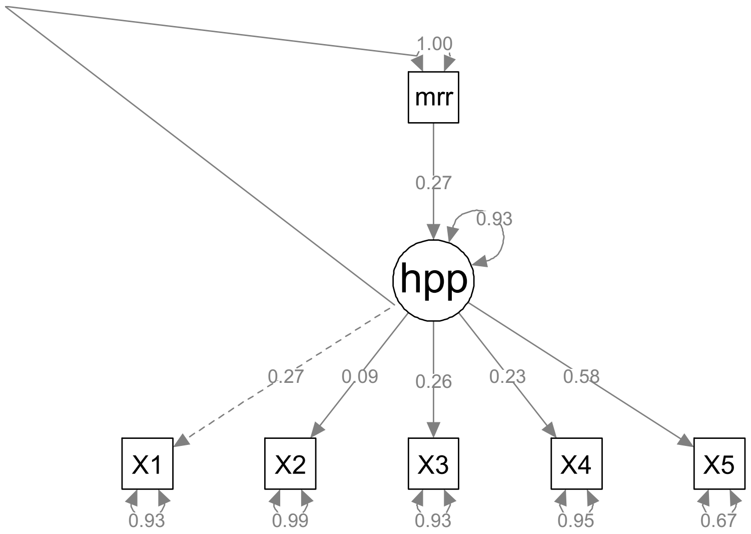


Structural Equation Model: Predicting Happiness Latent Variable

Measurement Model: Confirmatory Factor Analysis

Simultaneous Equations Implied by Path Diagram

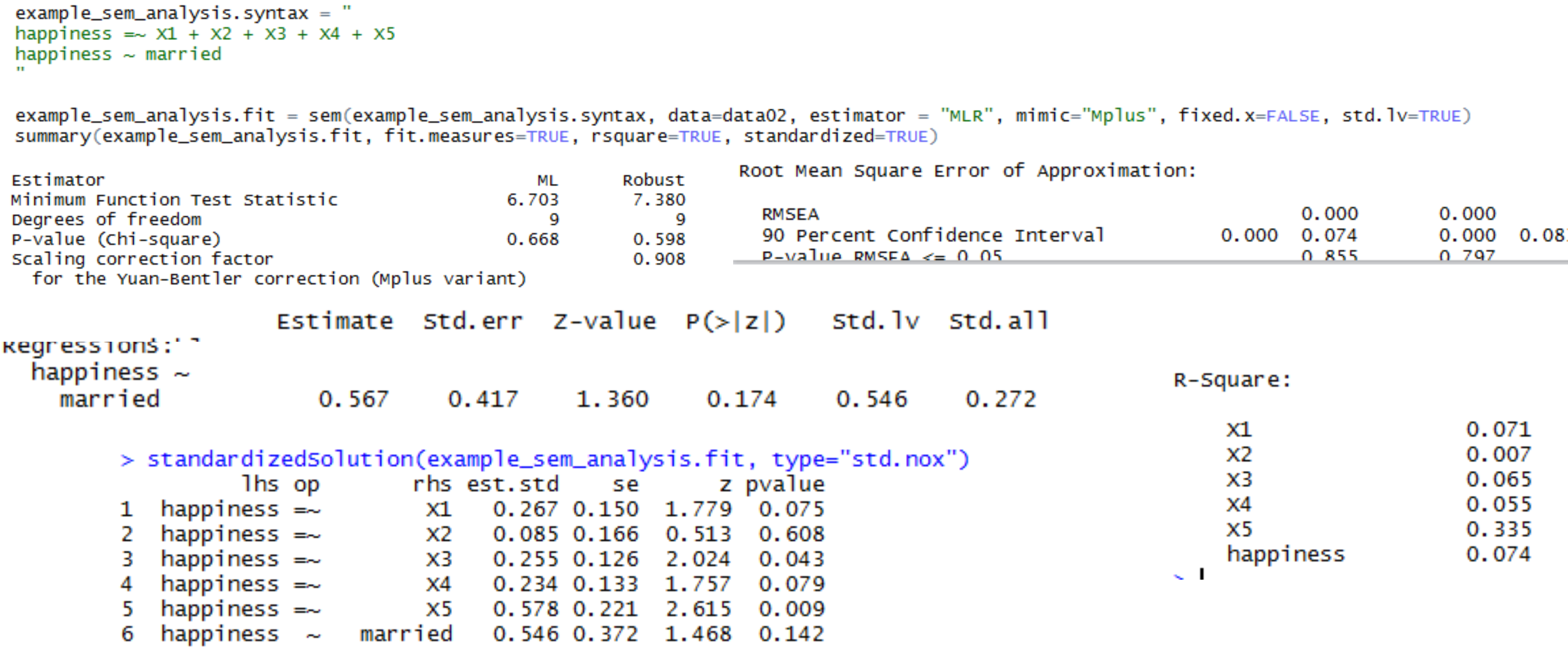
SEM is often called Path Analysis with Latent Variables



$$X_{p1} = \mu_1 + \lambda_1 HPP_p + e_{p1}$$
$$X_{p2} = \mu_2 + \lambda_2 HPP_p + e_{p2}$$
$$X_{p3} = \mu_3 + \lambda_3 HPP_p + e_{p3}$$
$$X_{p4} = \mu_4 + \lambda_4 HPP_p + e_{p4}$$
$$X_{p5} = \mu_5 + \lambda_5 HPP_p + e_{p5}$$
$$HPP_p = \beta_0 + \beta_1 Married_p + e_p^{HPP}$$

Example Data: SEM Analysis

- The SEM analysis (simultaneous) is the ideal: here is the syntax and the results



```
example_sem_analysis.syntax = "
happiness =~ X1 + X2 + X3 + X4 + X5
happiness ~ married
"

example_sem_analysis.fit = sem(example_sem_analysis.syntax, data=data02, estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)
summary(example_sem_analysis.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

```
Estimator                                         ML       Robust
Minimum Function Test Statistic                6.703        7.380
Degrees of freedom                                 9            9
P-value (Chi-square)                           0.668        0.598
Scaling correction factor                                   0.908
  for the Yuan-Bentler correction (Mplus variant)
```

```
Root Mean Square Error of Approximation:

  RMSEA                                          0.000       0.000
  90 Percent Confidence Interval          0.000  0.074       0.000  0.08
  P-value RMSEA <= 0.05                          0.855       0.797
```

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  happiness ~
    married           0.567    0.417    1.360    0.174    0.546    0.272
```

```
> standardizedSolution(example_sem_analysis.fit, type="std.nox")
        lhs op     rhs est.std    se     z pvalue
1 happiness =~      X1   0.267 0.150 1.779  0.075
2 happiness =~      X2   0.085 0.166 0.513  0.608
3 happiness =~      X3   0.255 0.126 2.024  0.043
4 happiness =~      X4   0.234 0.133 1.757  0.079
5 happiness =~      X5   0.578 0.221 2.615  0.009
6 happiness  ~ married   0.546 0.372 1.468  0.142
```

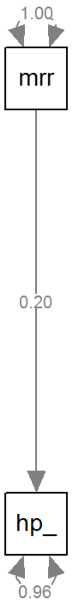
```
R-Square:

   X1               0.071
   X2               0.007
   X3               0.065
   X4               0.055
   X5               0.335
   happiness        0.074
```

Path Diagram of Same Analysis with Sum Score

- A common way of depicting SEMs is with a path diagram→ a pictorial representation of the statistical model
  - Observed variables: Squares
  - Latent variables: Circles
  - Direct effects: Arrows with one head
  - Indirect effects: Arrows with t

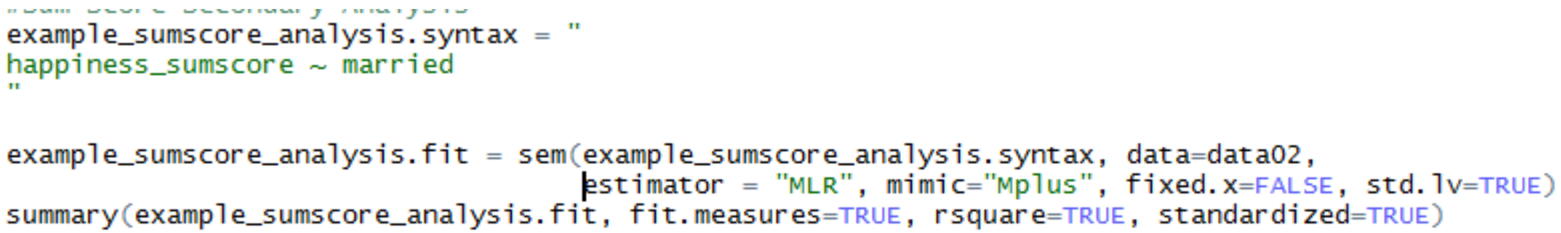
- From our previous GLM example

- Here MRR is marital status and hp_ is the happiness sum score



Same Analysis with Sum Score: Syntax and Results

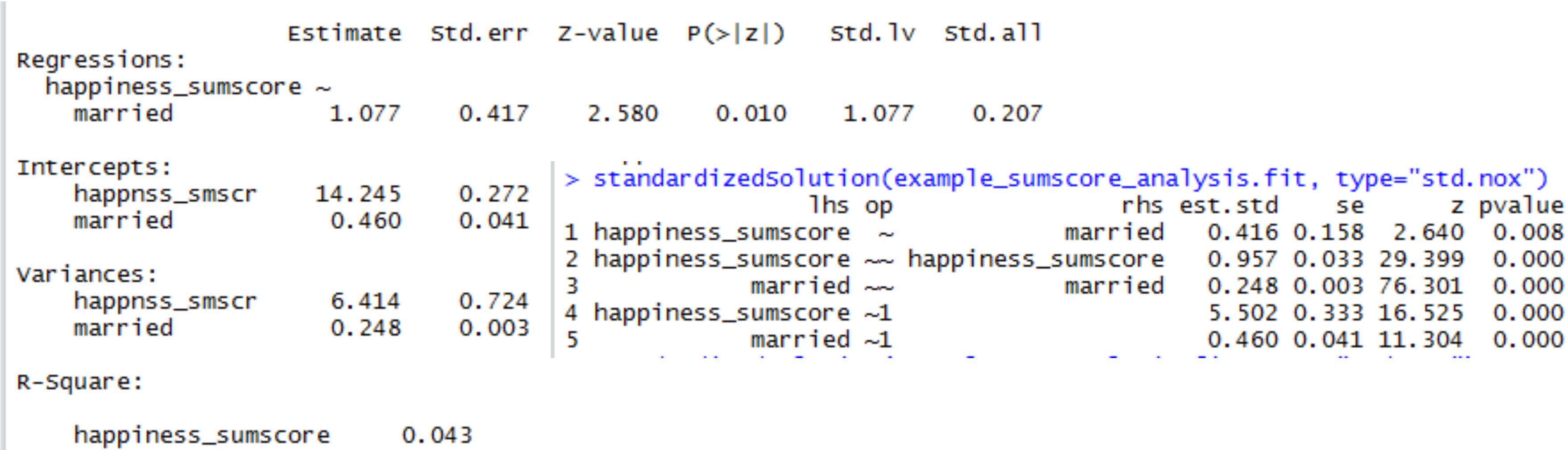
- The sum score analysis shows a different result:



```
#Sum Score Secondary Analysis
example_sumscore_analysis.syntax = "
happiness_sumscore ~ married
"

example_sumscore_analysis.fit = sem(example_sumscore_analysis.syntax, data=data02,
                                    estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)
summary(example_sumscore_analysis.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

Results:

Where is the model fit?



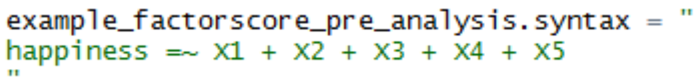
```
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  happiness_sumscore ~
    married           1.077    0.417    2.580    0.010    1.077    0.207

Intercepts:
    happnss_smscr    14.245    0.272
    married           0.460    0.041

Variances:
    happnss_smscr     6.414    0.724
    married           0.248    0.003

R-Square:

    happiness_sumscore      0.043
```

```
> standardizedSolution(example_sumscore_analysis.fit, type="std.nox")
                 lhs op                rhs est.std    se      z pvalue
1 happiness_sumscore  ~            married   0.416 0.158  2.640  0.008
2 happiness_sumscore ~~ happiness_sumscore   0.957 0.033 29.399  0.000
3            married ~~            married   0.248 0.003 76.301  0.000
4 happiness_sumscore ~1                      5.502 0.333 16.525  0.000
5            married ~1                      0.460 0.041 11.304  0.000
```

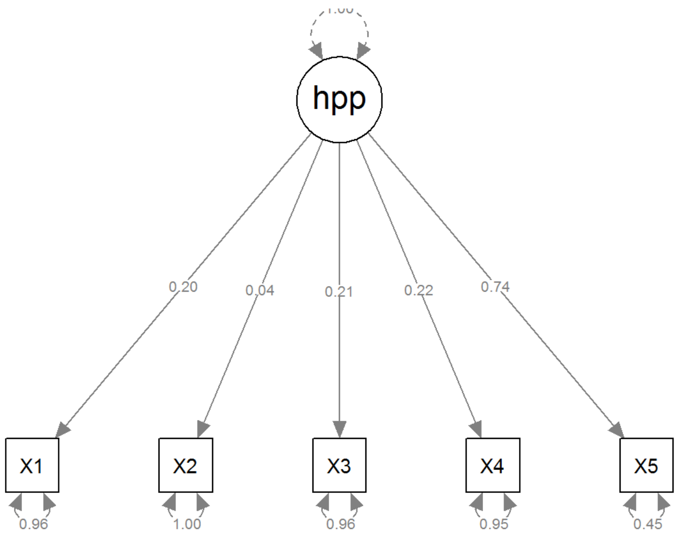
Analysis using a Factor Score

- To conduct the same analysis with a factor score instead of a sum score, there are two steps needed
  1. Run a CFA model only; check fit; obtain factor score estimate
  2. Run secondary analysis with factor score as observed variable

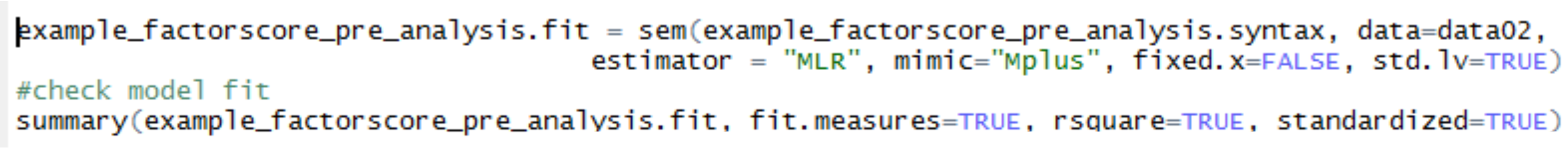
Step 1: Obtaining the factor score: use only the measurement model



```
example_factorscore_pre_analysis.syntax = "
happiness =~ X1 + X2 + X3 + X4 + X5
"
```

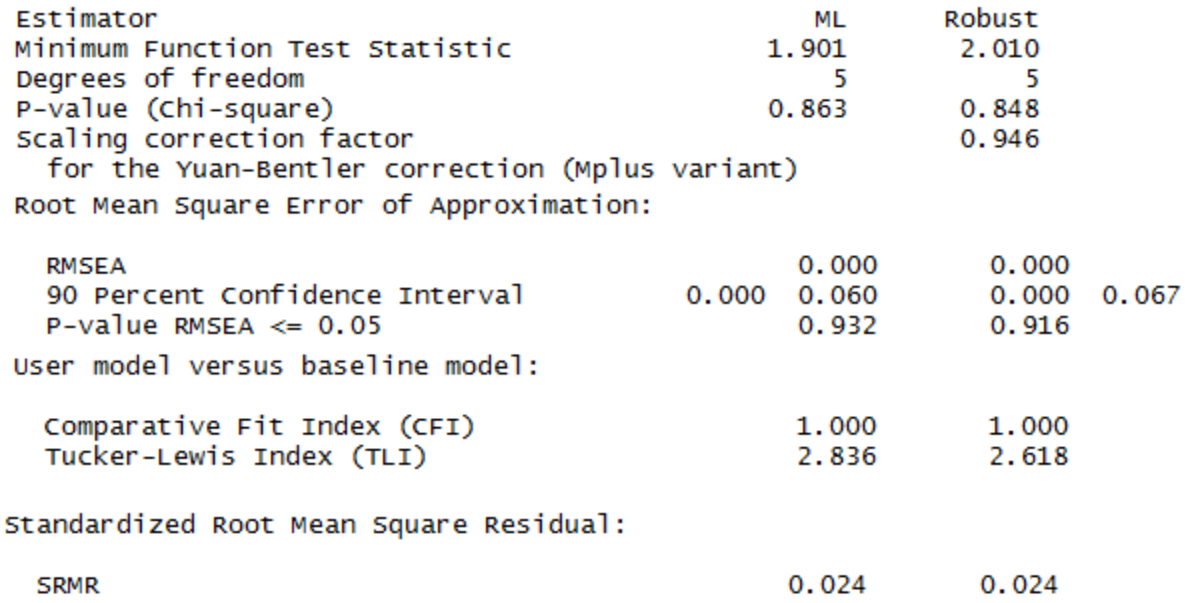


Obtaining the Factor Score: Checking for Model Fit



```
example_factorscore_pre_analysis.fit = sem(example_factorscore_pre_analysis.syntax, data=data02,
                                           estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)
#check model fit
summary(example_factorscore_pre_analysis.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
```

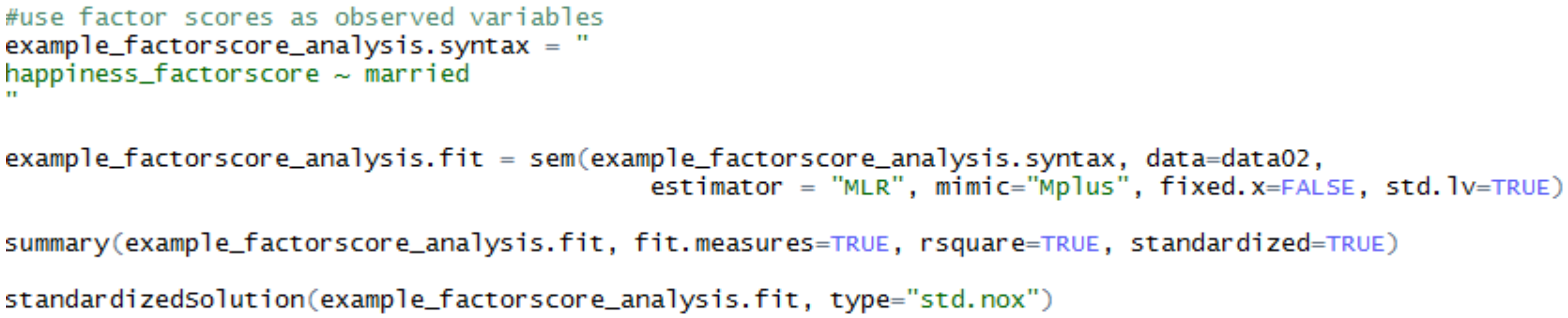
- Model fit:



```
Estimator                                         ML       Robust
Minimum Function Test Statistic                1.901        2.010
Degrees of freedom                                 5            5
P-value (Chi-square)                           0.863        0.848
Scaling correction factor                                   0.946
  for the Yuan-Bentler correction (Mplus variant)
Root Mean Square Error of Approximation:

  RMSEA                                         0.000        0.000
  90 Percent Confidence Interval         0.000  0.060        0.000  0.067
  P-value RMSEA <= 0.05                         0.932        0.916
User model versus baseline model:

  Comparative Fit Index (CFI)                   1.000        1.000
  Tucker-Lewis Index (TLI)                      2.836        2.618

Standardized Root Mean Square Residual:

  SRMR                                         0.024        0.024
```

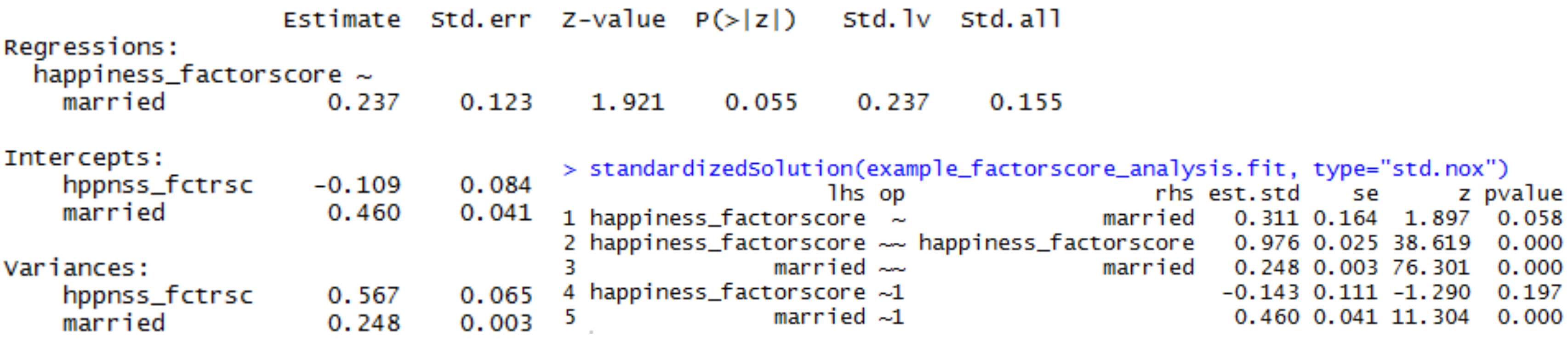
- Good model fit—lets now use the factor score

Using the Factor Score in the Analysis



```
#use factor scores as observed variables
example_factorscore_analysis.syntax = "
happiness_factorscore ~ married
"

example_factorscore_analysis.fit = sem(example_factorscore_analysis.syntax, data=data02,
                                       estimator = "MLR", mimic="Mplus", fixed.x=FALSE, std.lv=TRUE)

summary(example_factorscore_analysis.fit, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)

standardizedSolution(example_factorscore_analysis.fit, type="std.nox")
```

- Results:



```
                     Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  happiness_factorscore ~
    married             0.237    0.123    1.921    0.055    0.237    0.155

Intercepts:
    hppnss_fctrsc      -0.109    0.084
    married             0.460    0.041

Variances:
    hppnss_fctrsc       0.567    0.065
    married             0.248    0.003
```

```
> standardizedSolution(example_factorscore_analysis.fit, type="std.nox")
                    lhs op                   rhs est.std    se      z pvalue
1 happiness_factorscore  ~               married   0.311 0.164  1.897  0.058
2 happiness_factorscore ~~ happiness_factorscore   0.976 0.025 38.619  0.000
3               married ~~               married   0.248 0.003 76.301  0.000
4 happiness_factorscore ~1                        -0.143 0.111 -1.290  0.197
5               married ~1                         0.460 0.041 11.304  0.000
```

Side-by-Side Comparison

Model	Standardized Estimate (no.x)	Std Estimate Standard Error	Std Estimate p-value	Estimated R-Square
Simultaneous SEM	0.546	0.372	0.142	0.074
Sum Score	0.416	0.158	0.008	0.043
Factor Score	0.311	0.164	0.058	0.024

How To Incorporate Scores Into Secondary Analyses

How to Incorporate Scores into Secondary Analyses

- So far we have seen sum scores and factor scores and discussed their similarities and differences
- For secondary analyses:
  - Sum scores **by themselves** will not work because they do not provide any mechanism to detect for model misspecification and they ignore measurement error
    - Model misspecification error is likely much worse than any other type
  - Factor scores **by themselves** will not work because they ignore measurement error
    - Ensuring CFA model fit will help omit some misspecification error
- We will use factor scores as they are less prone to model misspecification error
  - But, we cannot use just one factor score as it will have measurement error present
- We will treat factor scores as missing data and multiply impute "plausible values" for multiple analyses with factor scores
  - e.g. Mislevy, Johnson, & Muraki (1992): Scaling procedures in NAEP

From a Missing Data Lecture: Bad Ways to Handle Missing Data

- Dealing with missing data is important, as the mechanisms you choose can dramatically alter your results

- This point was not fully realized when the first methods for missing data were created
  - Each of the methods described in this section should ***never be used***
  - Given to show perspective – and to allow you to understand what happens if you were to choose each

- If we think of the factor score (or true score from CTT) as being missing, then the use of a factor score or sum score is analogous to a single imputation

From a Missing Data Lecture: Single Imputation Methods

- **Single imputation** methods replace missing data with some type of value
  - **Single:** one value used
  - **Imputation:** replace missing data with value

- Upside: can use entire data set if missing values are replaced

- Downside: biased parameter estimates and standard errors (even if missing is MCAR)
  - Type-I error issues

- Still: never use these techniques

Why Single Imputation Is Bad Science

- Overall, the methods described in this section are not useful for handling missing data

- If you use them you will likely get a statistical answer that is an artifact
  - Actual estimates you interpret (parameter estimates) will be biased (in either direction)
  - Standard errors will be too small
    - Leads to Type-I Errors

- Putting this together: you will likely end up making conclusions about your data that are wrong

Multiple Imputation

Multiple Imputation

- Rather than using single imputation, a better method is to use multiple imputation
  - The multiply imputed values will end up adding variability to analyses – helping with biased parameter and SE estimates
- Multiple imputation is a mechanism by which you “fill in” your missing data with “plausible” values
  - End up with multiple data sets – need to run multiple analyses
  - Missing data are predicted using a statistical model using the observed data (the MAR assumption) for each observation
- MI is possible due to statistical assumptions
  - For CFA, we are helped by the fact that our data are multivariate normal

Multiple Imputation Steps

1. **Imputation:** The missing data are filled in a number of times (say, *m* times) to generate *m* complete data sets
   - For us, this is the factor score—drawn at random from each person's factor score distribution

2. **Analysis:** The *m* complete data sets are analyzed using standard statistical analyses
   - For **each data set** we then use lavaan like we normally would with the imputed factor score as an observed variable

3. **Results Pooling:** The results from the *m* complete data sets are combined to produce inferential results
   - We then combine each of our *m* analyses to produce the final analysis statistics from which we draw our conclusions and inferences

Distributions: The Key to Multiple Imputation

- The key idea behind multiple imputation is that each missing value has a **distribution** of likely values
  - The distribution reflects the uncertainty about what the variable may have been--this is pretty obvious for us as factor scores have distributions

- By contrast, single imputation **(using just the factor score or the sum score in an analysis)**, disregards the uncertainty in each missing data point
  - Results from singly imputed data sets may be biased or have higher Type-I errors

- Uncertainty == measurement error in when using scales

Example Imputation Analysis: Predicting Happiness (from our First Lecture)

Imputation Preliminary Information

- The first step is to create multiple data sets drawing a factor score for each person

- Recall the factor scores for each person (in CFA) follow a normal distribution with the mean and variance coming from model parameters:

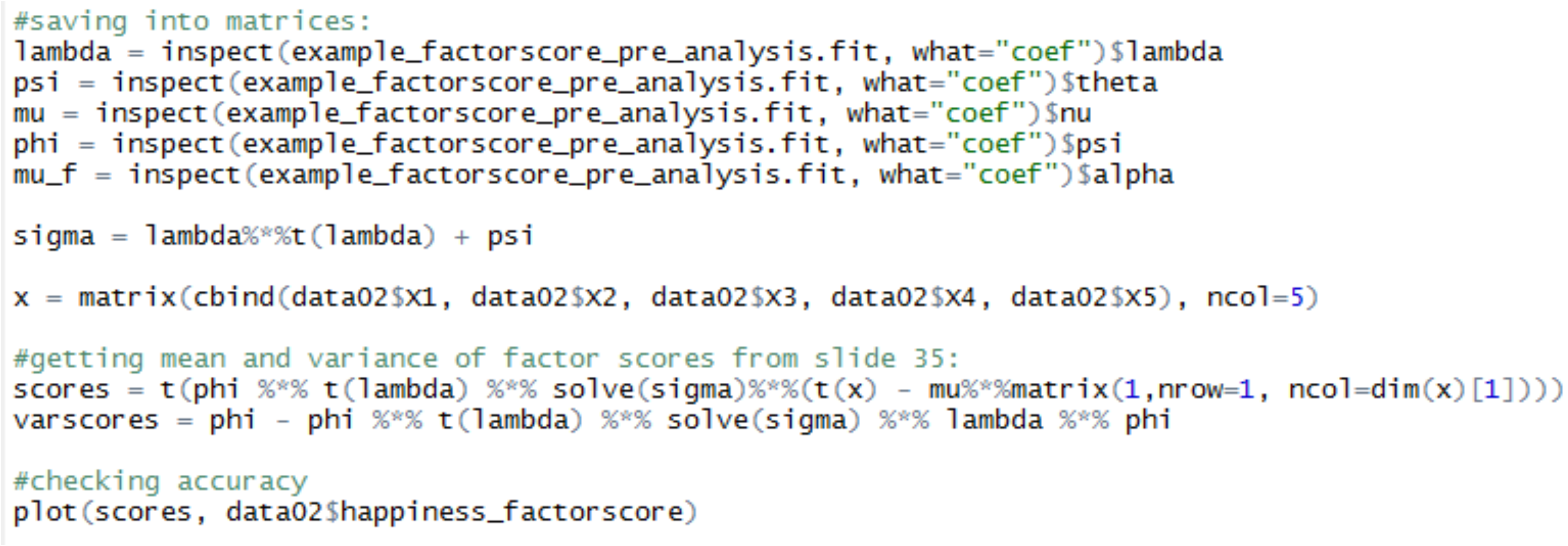
- Using the conditional distributions of MVNs result:

$f\left(\mathbf{F}_p \middle| \mathbf{Y}_p\right)$ is MVN:

With mean: $\boldsymbol{\mu}_F + \boldsymbol{\Phi}\boldsymbol{\Lambda}^T\left(\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1}\left(\mathbf{Y}_p^T - \boldsymbol{\mu}\right)$

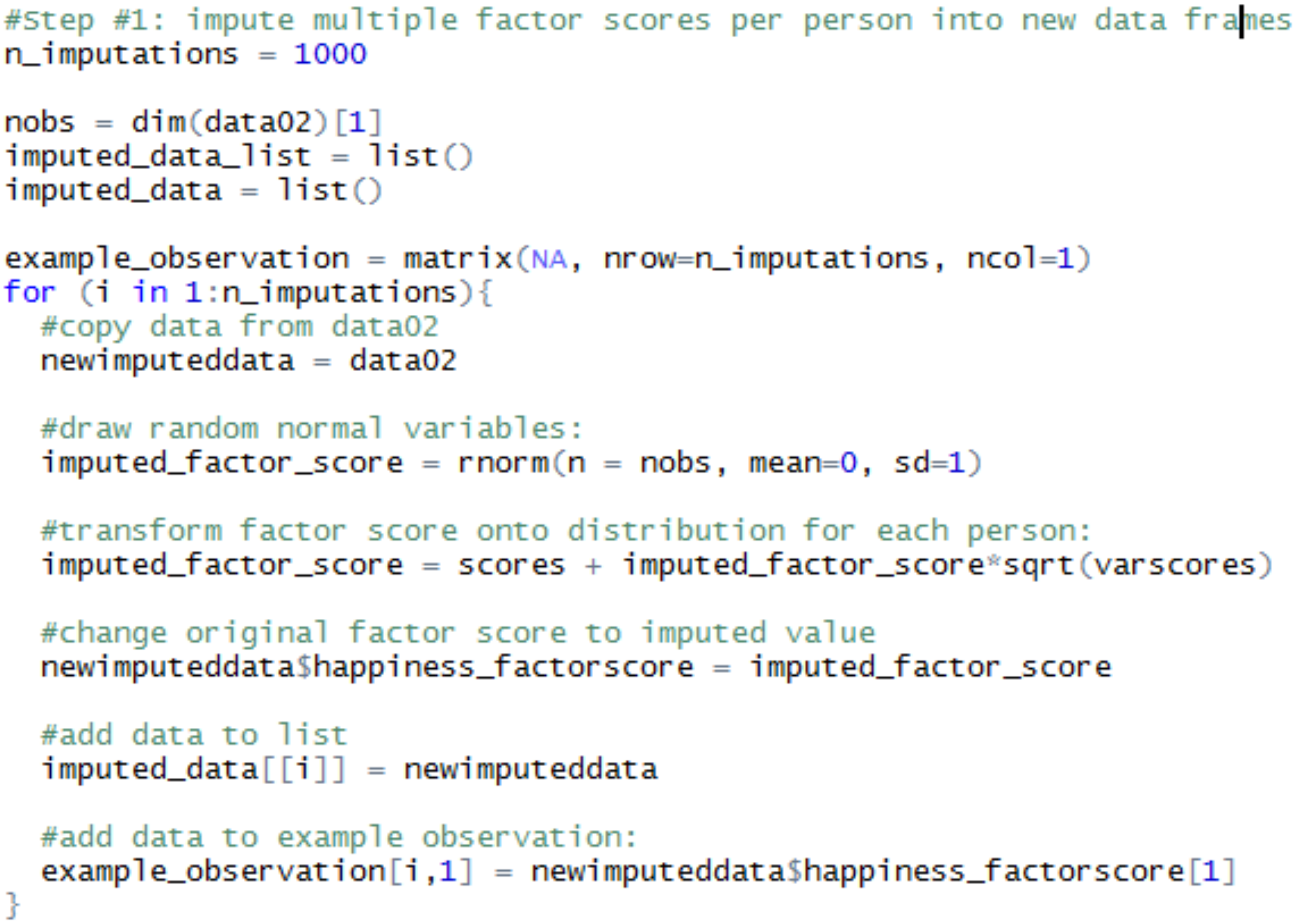
And Covariance: $\boldsymbol{\Phi} - \boldsymbol{\Phi}\boldsymbol{\Lambda}^T\left(\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}\right)^{-1}\boldsymbol{\Lambda}\boldsymbol{\Phi}$

#WTFTemplin



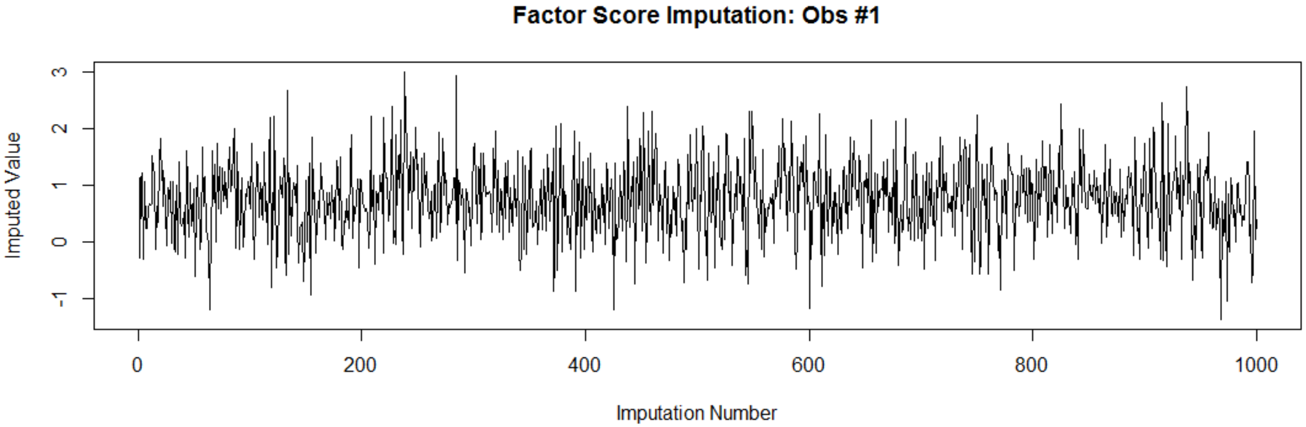
```r
#saving into matrices:
lambda = inspect(example_factorscore_pre_analysis.fit, what="coef")$lambda
psi = inspect(example_factorscore_pre_analysis.fit, what="coef")$theta
mu = inspect(example_factorscore_pre_analysis.fit, what="coef")$nu
phi = inspect(example_factorscore_pre_analysis.fit, what="coef")$psi
mu_f = inspect(example_factorscore_pre_analysis.fit, what="coef")$alpha

sigma = lambda%*%t(lambda) + psi

x = matrix(cbind(data02$X1, data02$X2, data02$X3, data02$X4, data02$X5), ncol=5)

#getting mean and variance of factor scores from slide 35:
scores = t(phi %*% t(lambda) %*% solve(sigma)%*%(t(x) - mu%*%matrix(1,nrow=1, ncol=dim(x)[1])))
varscores = phi - phi %*% t(lambda) %*% solve(sigma) %*% lambda %*% phi

#checking accuracy
plot(scores, data02$happiness_factorscore)
```

Step #1: Generate Multiple Data Sets of

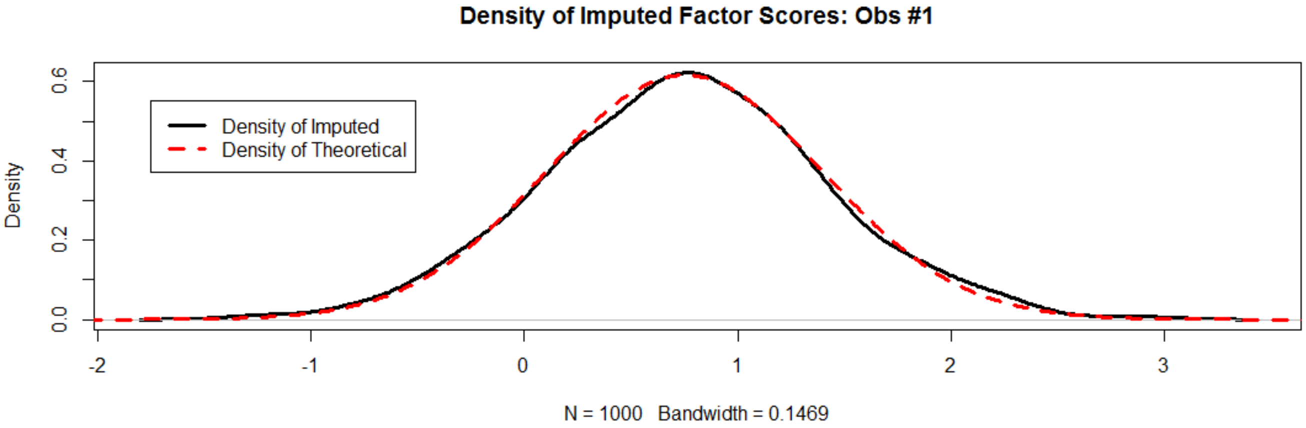


```
#Step #1: impute multiple factor scores per person into new data frames
n_imputations = 1000

nobs = dim(data02)[1]
imputed_data_list = list()
imputed_data = list()

example_observation = matrix(NA, nrow=n_imputations, ncol=1)
for (i in 1:n_imputations){
  #copy data from data02
  newimputeddata = data02

  #draw random normal variables:
  imputed_factor_score = rnorm(n = nobs, mean=0, sd=1)

  #transform factor score onto distribution for each person:
  imputed_factor_score = scores + imputed_factor_score*sqrt(varscores)

  #change original factor score to imputed value
  newimputeddata$happiness_factorscore = imputed_factor_score

  #add data to list
  imputed_data[[i]] = newimputeddata

  #add data to example observation:
  example_observation[i,1] = newimputeddata$happiness_factorscore[1]
}
```

Sequence of Imputed Factor Scores for Observation #1



Factor Score Imputation: Obs #1



Density of Imputed Factor Scores: Obs #1

Pooling Parameters from Analyses of Imputed Data Sets

- In the pooling phase, the results are pooled and reported

- For parameter estimates, the pooling is straight forward
  - The estimated parameter is the average parameter value across all imputed data sets
    - For our example the average slope comes from the average slope of all 1000 analyses

- For standard errors, pooling is more complicated
  - Have to worry about sources of variation:
    - Variation from sampling error that would have been present had the data not been missing
    - Variation from sampling error resulting from missing data

Pooling Standard Errors Across Imputation Analyses

- Standard error information comes from two sources of variation from imputation analyses (for m imputations)

- Within Imputation Variation:

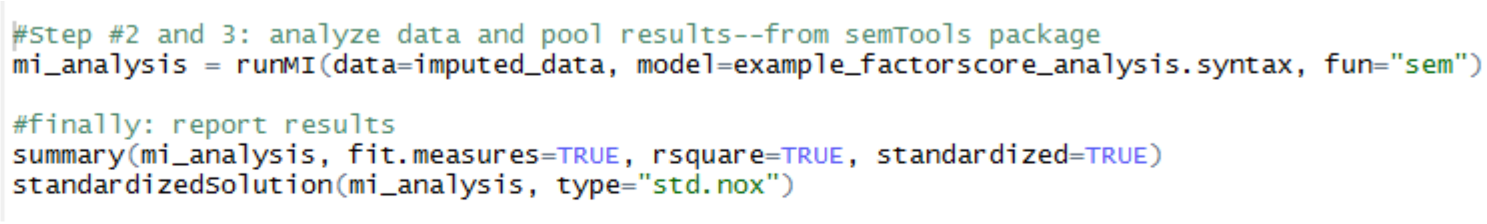
$$V_W = \frac{1}{m}\sum_{i=1}^{m} SE_i^2$$

- Between Imputation Variation (here θ is an estimated parameter from an imputation analysis):

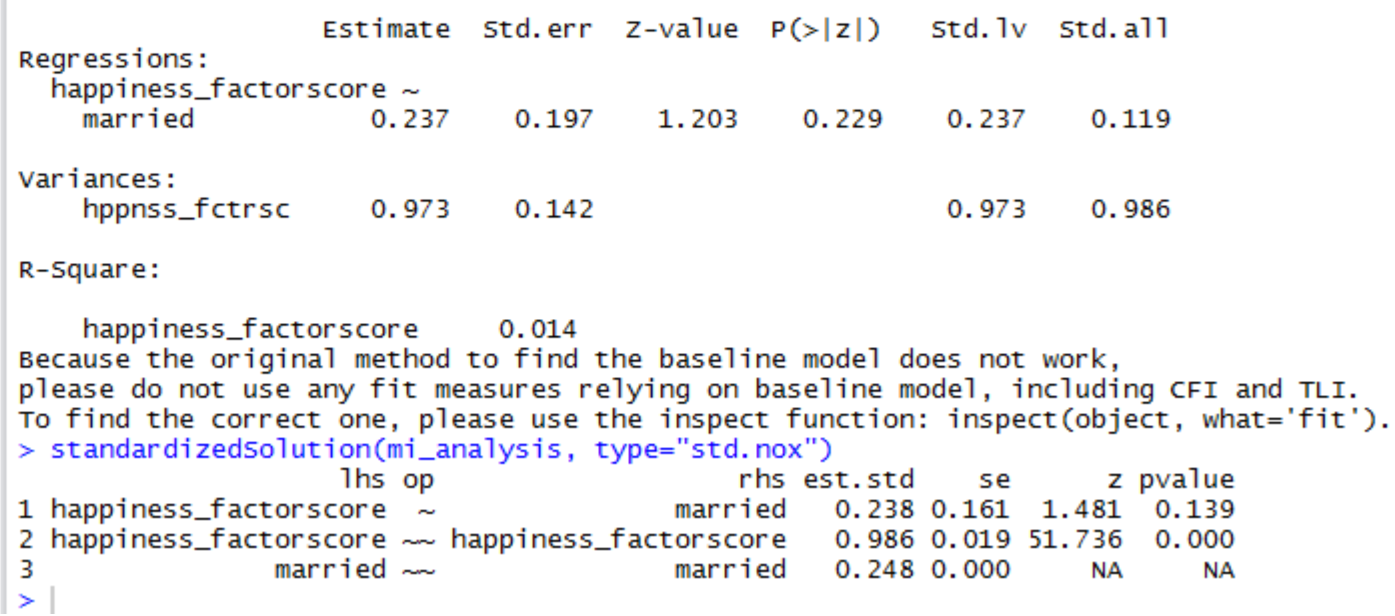
$$V_B = \frac{1}{m-1}\sum_{i=1}^{m}\left(\hat{\theta}_i - \bar{\theta}\right)^2$$

- Then, the total sampling variance is: $V_T = V_W + V_B + \frac{V_B}{M}$
- The subsequent (imputation pooled) SE is $SE = \sqrt{V_T}$

Step #2 (Analysis) and Step #3 (Pooling)



```
#Step #2 and 3: analyze data and pool results--from semTools package
mi_analysis = runMI(data=imputed_data, model=example_factorscore_analysis.syntax, fun="sem")

#finally: report results
summary(mi_analysis, fit.measures=TRUE, rsquare=TRUE, standardized=TRUE)
standardizedSolution(mi_analysis, type="std.nox")
```



```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  happiness_factorscore ~
    married            0.237    0.197    1.203    0.229    0.237    0.119

Variances:
    hppnss_fctrsc      0.973    0.142                      0.973    0.986

R-Square:

    happiness_factorscore    0.014
Because the original method to find the baseline model does not work,
please do not use any fit measures relying on baseline model, including CFI and TLI.
To find the correct one, please use the inspect function: inspect(object, what='fit').
> standardizedSolution(mi_analysis, type="std.nox")
                    lhs op                   rhs est.std    se      z pvalue
1 happiness_factorscore  ~               married   0.238 0.161  1.481  0.139
2 happiness_factorscore ~~ happiness_factorscore   0.986 0.019 51.736  0.000
3               married ~~               married   0.248 0.000     NA     NA
> |
```

Side-by-Side Comparison

Model	Standardized Estimate (no.x)	Std Estimate Standard Error	Std Estimate p-value	Estimated R-Square
Simultaneous SEM	0.546	0.372	0.142	0.074
Sum Score	0.416	0.158	0.008	0.043
Factor Score	0.311	0.164	0.058	0.024
Factor Score Imputation	0.238	0.161	0.139	0.014

Error Score Card

Error Source	Sum Score	Factor Score	Factor Score Imputation
Measurement Error	🟥	🟥	🟩
Sampling Error	🟩	🟥	🟥
Prior Distribution Bias	🟩	🟥	🟥
ML Estimator Bias	🟩	🟥	🟥
MSPFD: Dimensionality	🟥	🟩	🟩
MSPFD: Constraints	🟥	🟩	🟩
MSPFD: Lin Pred Function	🟥	🟩	🟩
MSPFD: Data Distribution	🟥	🟥	🟥
MSPFD: Factor Distribution	🟥	🟥	🟥

Red: Source of Error Potentially Present **Green: Not a Source of Error**

Don’t count red-vs-green as some types of error is worse than others

Ways To Reduce Error in Secondary Analyses

Ways to Reduce Error in Secondary Analyses

- For each source of error in a secondary analysis there are ways to reduce that error so that secondary analyses are able to be completed with a good degree of accuracy

- Some of the ways to do so are very difficult if not impossible with current methods…Some are very possible

- All of the following slides assume that **no sum-score analysis is used: only factor score-based analyses with imputation**

- This section outlines each source of error and how to reduce such error

Ways to Reduce the Impact of Measurement Error

- **To reduce measurement error:**
  - Have a larger number of high-quality and highly informative items measure each factor

- **Ramifications of reducing this type of error:**
  - Greater reliability for the factor score/lessened measurement error
  - Greater power
  - Less need for large number of imputation steps

- **Statistical methods needed if error is present:**
  - Multiple imputation of plausible values of factor scores using factor score distribution from measurement model-only analysis

- **Difficulties in error reduction approach above:**
  - More items makes achieving model fit much more difficult

Ways to Reduce the Impact of Prior Factor Score Distribution Error

- **To reduce prior factor score distribution error:**
  - Have a larger number of high-quality and highly informative items measure each factor

- **Ramifications of reducing this type of error:**
  - More items makes the information from the data overwhelm the information from the prior distribution
  - Greater reliability for the factor score/lessened measurement error
  - Greater power
  - Less need for large number of imputation steps

- **Statistical methods needed if this type of error is present:**
  - Multiple imputation of plausible values of factor scores using factor score distribution from measurement model-only analysis

- **Difficulties in error reduction approach above:**
  - More items makes achieving model fit much more difficult

Ways to Reduce the Impact of Sampling Error

- **To reduce sampling error:**
  - Have a large sample size

- **Ramifications of reducing this type of error:**
  - Greater power for testing hypotheses
  - Greater stability of factor score distributions
  - Less need for large number of imputation steps

- **Statistical methods needed if this type of error is present:**
  - **Multiple imputation of plausible values of factor scores** using **multiply imputed** factor score distribution from measurement model-only analysis
  - All can be accomplished in an MCMC analysis where all parameters are estimated simultaneously with factor scores

- **Difficulties in error reduction approach above:**
  - Hard to collect sample

Ways to Reduce the Impact ML Estimator Bias Error

- **To reduce ML Estimator Bias error:**

Have a large sample size – and –

Use an algorithm that uses the distribution of the residuals rather than the data (e.g., Residual ML vs. ML—but in a Bayesian context when imputing factor scores)

- **Ramifications of reducing this type of error:**
  - More accurate estimates of factor score distributions
  - Better Type-I error rate prevention in small sample sizes
- **Statistical methods needed if this type of error is present:**
  - Analysis algorithms with REML-based distributions
- **Difficulties in error reduction approach above:**
  - Very few exist for CFA
  - Existing algorithms often not able to provide all information needed for analyses

Ways to Reduce the Impact Model Misspecification Error of Types:
Dimensionality, Constraints, and Linear Predictor Function

- **To reduce model misspecification error due to dimensionality, parameter constraints, and linear predictor function:**

Achieve good model fit in your measurement model

- **Ramifications of reducing this type of error:**
  - More accurate estimates of factor score distributions
  - Better Type-I error rate prevention in small sample sizes
  - Better Type-II error rate prevention
  - More accurate results

- **Statistical methods needed if this type of error is present:**
  - Any analysis algorithm with indications of goodness of model fit

- **Difficulties in error reduction approach above:**
  - Harder to get model fit in Bayesian methods—and with non-normal data distributions

Ways to Reduce the Impact Model Misspecification Error of Due to Data Distributional Assumption Error

- **To reduce model misspecification error due to data distributional assumptions:**

Estimate your measurement model using multiple assumed distributions then compare model fit using methods like the Vuong test (Vuong, 1989; Merkle, You, & Preacher, under review: http://arxiv.org/pdf/1402.6720.pdf)

- **Ramifications of reducing this type of error:**
  - More accurate estimates of factor score distributions
  - Better Type-I error rate prevention in small sample sizes
  - Better Type-II error rate prevention
  - Much more accurate results

- **Statistical methods needed if this type of error is present:**
  - Estimators for multiple types of data and post-estimator model comparisons

- **Difficulties in error reduction approach above:**
  - Methods only exist for a handful (if any);
  - To the best of my knowledge, not currently possible without developing your own software

Ways to Reduce the Impact Model Misspecification Error of Due to Factor Distributional Assumption Error

- **To reduce model misspecification error due to factor distributional assumptions:**

Estimate multiple measurement models using multiple assumed factor distributions and multiple assumed data distributions then compare model fit using methods like the Vuong test (Vuong, 1989)

- **Ramifications of reducing this type of error:**
  - More accurate estimates of factor score distributions
  - Better Type-I error rate prevention in small sample sizes
  - Better Type-II error rate prevention
  - Much more accurate results

- **Statistical methods needed if this type of error is present:**
  - Estimators for multiple types of data, multiple types of factor distributions, and post-estimator model comparisons

- **Difficulties in error reduction approach above:**
  - To the best of my knowledge, not currently possible without developing your own software

Wrapping Up

Wrapping Up

- The use of scores is ubiquitous in research
  - Almost no research does what I am suggesting be done

- To use scores properly, be sure to understand
  - What a score represents
  - What sources of error are present within scores

- These are the most important lectures this semester
  - It provides a WHY and HOW for doing science with errors of measurement
  - Take this information and compare it to how people use scores…like: http://www.ets.org/s/gre/pdf/gre_guide.pdf

- The use of methods like these will improve your research (not to mention any operational psychometrics) by eliminating results that are purely due to chance
  - You will not be chasing what very well may be noisy results
  - Good topic discussion (different context-multiple comparisons but still valid here): http://www.stat.columbia.edu/~gelman/research/unpublished/p_hacking.pdf