

# DM25 SLIDES (Testlet Models, Version 1.0)

## 1. Module Overview

### 1.1 Module Cover (START)



### 1.2 Instructors

A screenshot of a 'Meet the instructors' section. It features two video player cards. The left card shows a woman with glasses and the name 'Hong Jiao' with 'University of Maryland, College Park' below it. The right card shows a woman and the name 'Manqian Liao' with 'Duolingo' below it. Both cards have a play button icon at the bottom right. A 'Back' button is located at the bottom right of the card area.

### **1.3 Designers**

Meet the designers:

The interface shows two video player cards side-by-side. Each card features a portrait of a man, a play button icon at the bottom right, and a caption below it. The left card is for 'Jon Lehrfeld' from 'ETS'. The right card is for 'André A. Rupp' from 'Mindful Measurement'.

Jon Lehrfeld  
ETS

André A. Rupp  
Mindful Measurement

Back

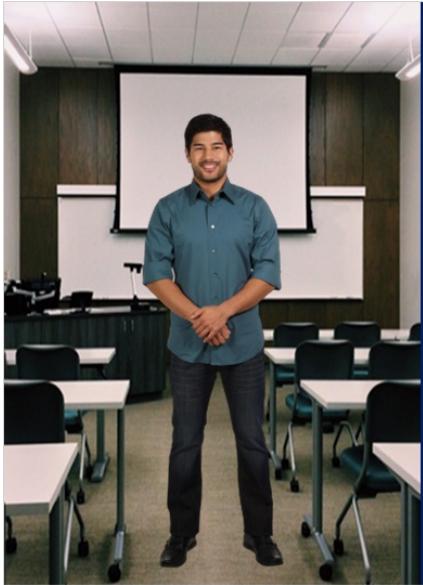
### **1.4 Welcome**

Welcome to the  
ITEMS Module!

The man to the left is Jet!  
Along with the instructors  
she will be guiding you  
through the module content.

Please enter your name below:

## Untitled Layer 1 (Slide Layer)

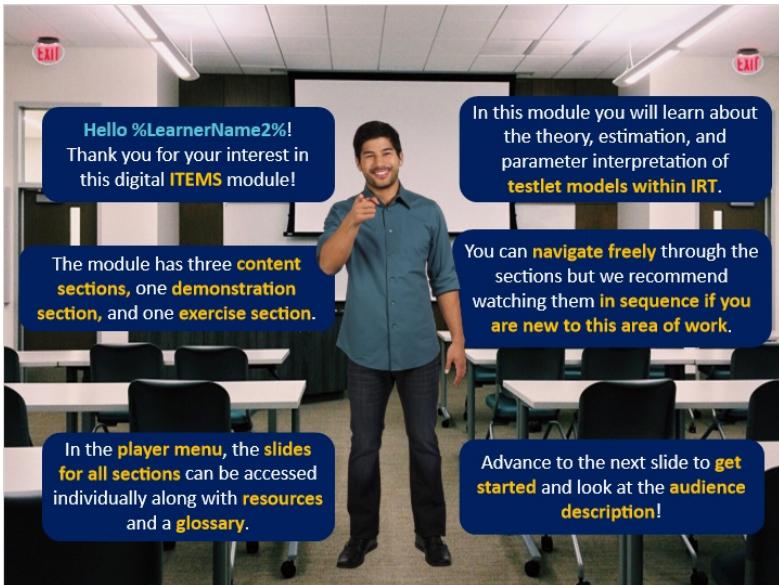


Welcome to the  
ITEMS Module!

The man to the left is Jet!  
Along with the instructors  
she will be guiding you  
through the module content.

Please enter your name below:

### 1.5 Overview



Hello %LearnerName2%!  
Thank you for your interest in  
this digital ITEMS module!

In this module you will learn about  
the theory, estimation, and  
parameter interpretation of  
**testlet models within IRT**.

The module has three **content sections**, one **demonstration section**, and one **exercise section**.

You can **navigate freely** through the  
sections but we recommend  
watching them **in sequence if you**  
**are new to this area of work**.

In the **player menu**, the **slides for all sections** can be accessed  
individually along with **resources**  
and a **glossary**.

Advance to the next slide to **get started** and look at the **audience description**!

## 1.6 Target Audience

**Target Audience**

Anyone who would like a gentle statistical introduction to this topic:

- graduate students and faculty in Master's, Ph.D., or certificate programs
- psychometricians and other measurement professionals
- data scientists / analysts
- research assistants or research scientists
- technical project directors
- assessment developers



However, we hope that you find the information in this module useful no matter what your official title or role in an organization is!

## 1.7 Expectations (I)



**Let's discuss expectations....**

## 1.8 Expectations (II)

**ITEMS Modules in Context**

The image displays a composite of two items. On the left is the front cover of the book "Testlet Response Theory and Its Applications" by Howard Wainer, Eric T. Bradlow, and Xiaohui Wang, published by Cambridge University Press. On the right is a screenshot of a university website for the Department of Human Development and Quantitative Methodology (HDQM) at the College of Education. Both the book cover and the website page are heavily redacted with large red X's.

## 1.9 Learning Objectives

**Learning Objectives**

The image features a central graphic of a target with an arrow hitting the bullseye, symbolizing achievement or success. Below the target are six numbered learning objectives listed in boxes:

1. Describe key components of testlet response theory
2. Understand key perspectives to conceptualize a testlet effect
3. Apply testlet response theory to scale construction
4. Perform testlet response model analysis using selected computer programs
5. Interpret the nature of testlet effects and the associated model parameters
6. Understand strategies for developing new testlet models

## **1.10 Prerequisites**

**Prerequisites**



1. Foundations of unidimensional item response theory
2. Basic mathematical transformation functions (e.g., logarithms)
3. Foundations of Bayesian estimation

## **1.11 Module Citation**

**Module Citation**

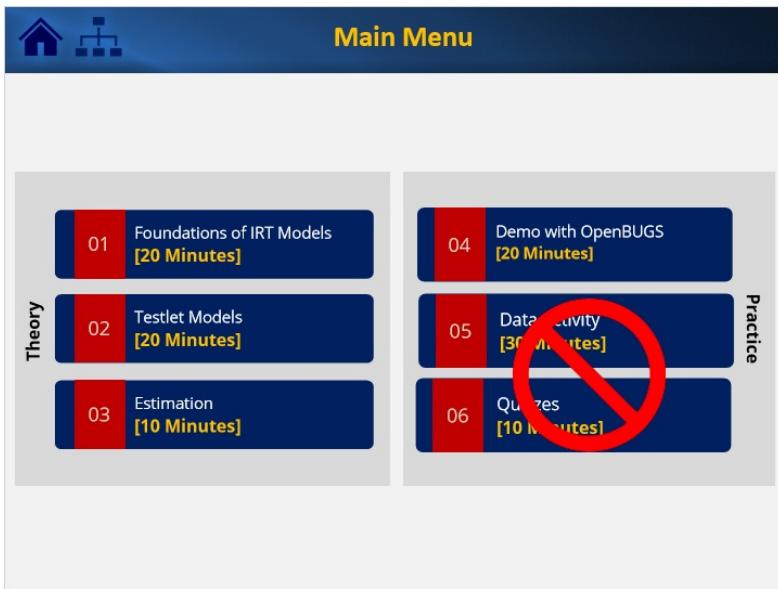
**Module Citation**

Jiao, H., & Liao, M. (2021). Testlet response models (Digital ITEMS Module 25).  
*Educational Measurement: Issues and Practice*, 40(3).



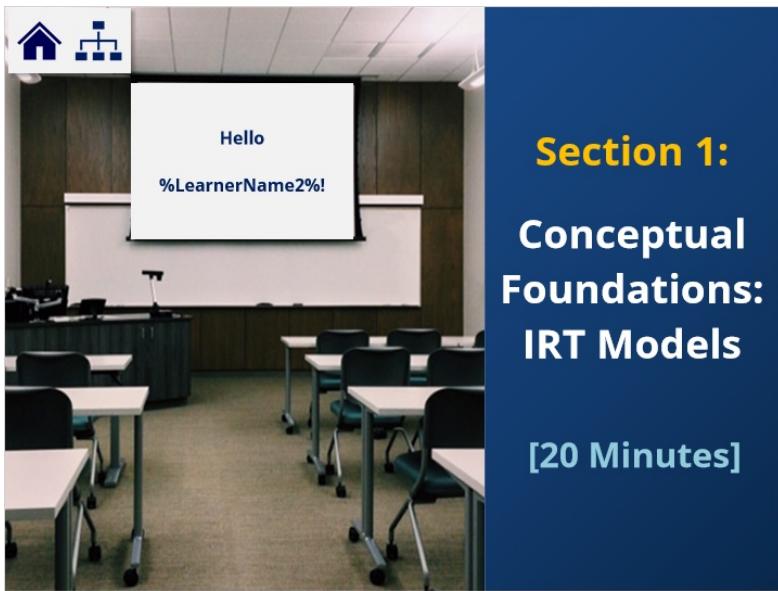
**FREE WEB RESOURCES**

## 1.12 Main Menu



## 2. Section 1: Conceptual Foundations: IRT Models

### 2.1 Cover: Section 1



## 2.2 Learning Objectives: Section 1

### Learning Objectives



1. Understand basic dichotomous IRT models  
2. Understand basic polytomous IRT models  
3. Understand model assumptions

## 2.3 Conceptual Foundations-Item Response Theory

### Conceptual Foundations: Item Response Theory

- Item response theory (IRT)
  - ✓ Dichotomous IRT models
  - ✓ Polytomous IRT models
- Logit vs. probit IRT models
- Hierarchical IRT model
- Assumptions for IRT models



## 2.4 Item Response Theory

**Item Response Theory (IRT)**

- A modern measurement framework that quantifies the relationship among an item response and a person's latent ability given item characteristics.

Item characteristic curves

adapted from <https://www.stata.com/features/overview/irt/>

- A mathematical representation modeling the nonlinear relationship between the probability of an item response, a latent ability level, and item parameters.

## 2.5 Item Response Theory Models

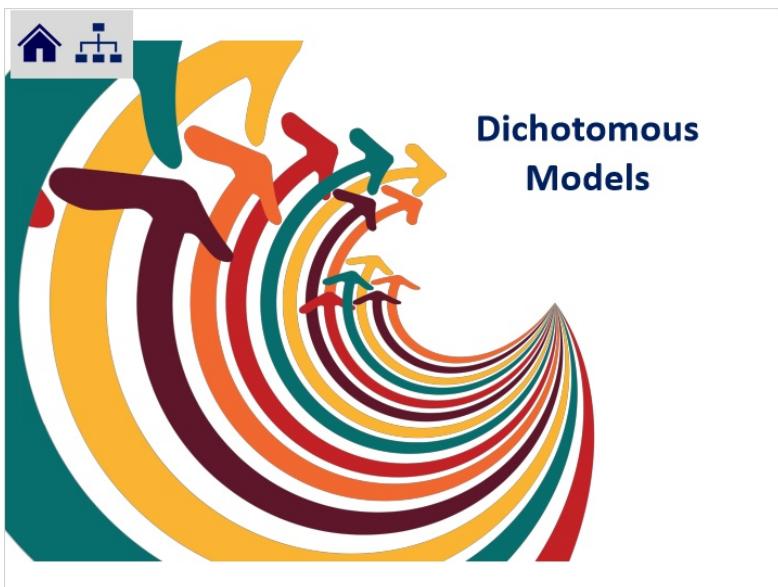
**IRT Models**

- The common IRT models include models for dichotomous responses and polytomous responses.
- Common IRT models for dichotomous item responses:
  - ✓ the Rasch measurement model
  - ✓ the two-parameter IRT model
  - ✓ the three-parameter IRT model
- Common IRT models for polytomous item responses:
  - ✓ the partial credit model
  - ✓ the generalized partial credit model
  - ✓ the graded response model

## 2.6 Topic Selection



## 2.7 Bookmark: Dichotomous Models



## 2.8 The Rasch Measurement Model I

The Rasch Measurement Model

- A **latent logistic model** modeling the **nonlinear relationship** between the probability of an item response and a latent ability level given an item difficulty parameter
- Such a nonlinear relationship can be graphically represented in an **item characteristic curve** (ICC) to show how an increase in latent ability leads to an increase in the probability of a correct item response given the specific item difficulty
- The **logit of a correct response** is linearly related to the interaction between a person's latent ability,  $\theta$ , and the item difficulty,  $b$

## 2.9 The Rasch Measurement Model II

The Rasch Measurement Model

**Logistic model**

$$p_{ij}(x_{ij} = 1|\theta_j) = \frac{\exp(\theta_j - b_i)}{1 + \exp(\theta_j - b_i)}$$
$$p_{ij}(x_{ij} = 1|\theta_j) = \frac{1}{1 + \exp[-(\theta_j - b_i)]}$$

**Item Characteristic Curve**

**Logit format**

$$\ln\left(\frac{P_{ij}}{1 - P_{ij}}\right) = (\theta_j - b_i)$$

## 2.10 The Two-Parameter Logistic Model

  The Two-Parameter Logistic Model

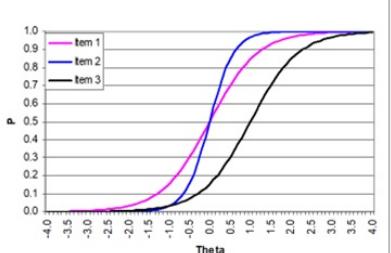
**Logistic model**

$$p_{ij}(x_{ij} = 1|\theta_j) = \frac{1}{1 + \exp[-a_i(\theta_j - b_i)]}$$

**Logit format**

$$\ln\left(\frac{P_{ij}}{1 - P_{ij}}\right) = a_i(\theta_j - b_i)$$

**Item Characteristic Curve**



The graph shows three sigmoidal curves representing Item Characteristic Curves for three different items. The x-axis is labeled 'Theta' and ranges from -4.0 to 4.0. The y-axis is labeled 'P' and ranges from 0.0 to 1.0. Item 1 (magenta) has the steepest slope and crosses the 0.5 probability mark at approximately Theta = 0.5. Item 2 (blue) follows Item 1 closely but is slightly less steep. Item 3 (black) has the shallowest slope and crosses the 0.5 probability mark at approximately Theta = 1.2.

## 2.11 The Three-Parameter Logistic Model

  The Three-Parameter Logistic Model

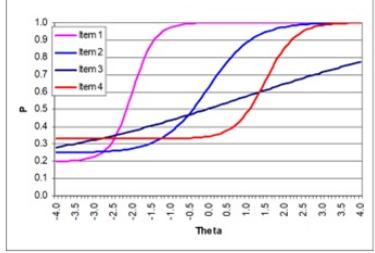
**Logistic model**

$$p_{ij}(x_{ij} = 1|\theta_j) = c_i + \frac{1 - c_i}{1 + \exp[-a_i(\theta_j - b_i)]}$$

**Logit format**

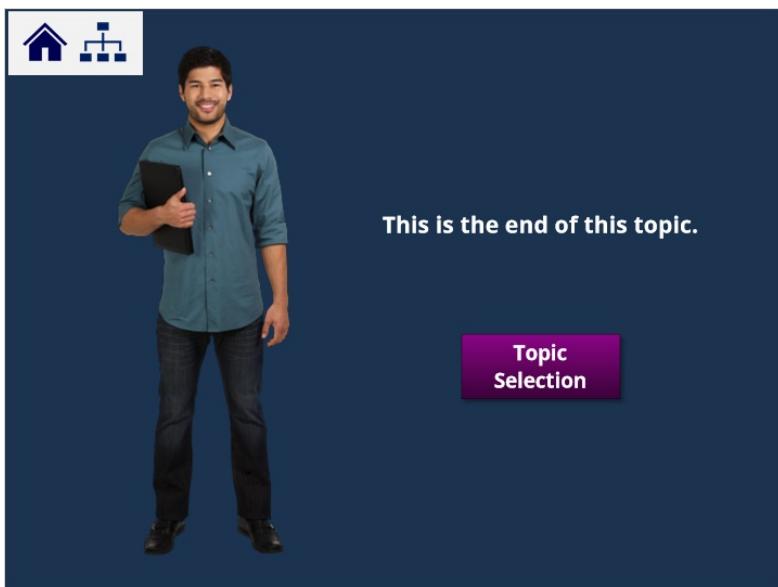
$$\ln\left(\frac{P_{ij} - c_i}{1 - P_{ij}}\right) = a_i(\theta_j - b_i)$$

**Item Characteristic Curve**

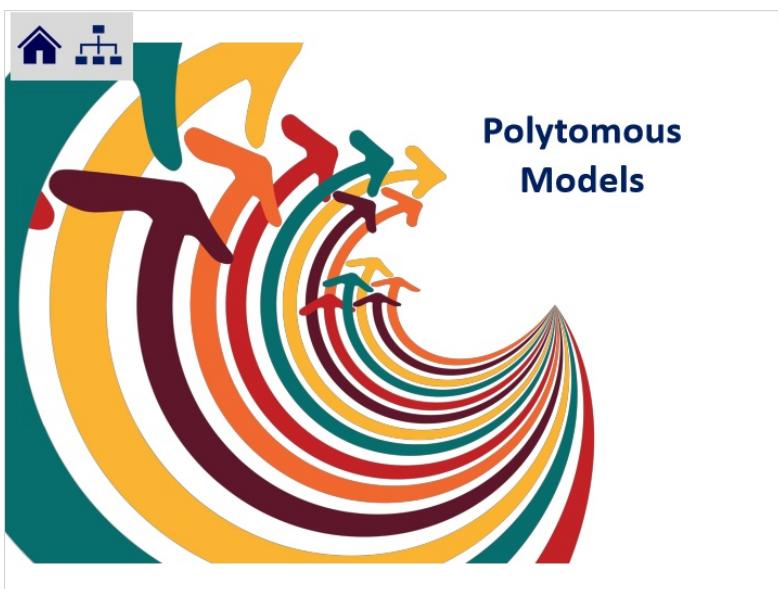


The graph shows four sigmoidal curves representing Item Characteristic Curves for four different items. The x-axis is labeled 'Theta' and ranges from -4.0 to 4.0. The y-axis is labeled 'P' and ranges from 0.0 to 1.0. Item 1 (magenta) has the steepest slope and crosses the 0.5 probability mark at approximately Theta = 0.5. Item 2 (blue) follows Item 1 closely but is slightly less steep. Item 3 (dark blue) has a moderate slope and crosses the 0.5 probability mark at approximately Theta = 1.2. Item 4 (red) has the shallowest slope and crosses the 0.5 probability mark at approximately Theta = 2.0.

## **2.12 Bookend: Dichotomous Models**



## **2.13 Bookmark: Polytomous Models**



## 2.14 Overview of Different Polytomous Models



### Overview of Polytomous Models

**Adjacent-category models**

Graded response model (e.g., Samejima, 1969)  
Modified graded response model (e.g., Muraki, 1990)

**Divide-by-total models**

Partial credit model (e.g., Masters, 1982)  
Generalized partial credit model (e.g., Muraki, 1992)  
Rating scale model (e.g., Andrich, 1978)  
Modified rating scale model (e.g., du Toit et al., 1999)  
Nominal response model (e.g., Bock, 1972)

## 2.15 The Graded Response Model I



### The Graded Response Model

**2PL IRT model**

$$p_{ij}(x_{ij} = 1|\theta_j) = \frac{\exp[a_i(\theta_j - b_i)]}{1 + \exp[a_i(\theta_j - b_i)]}$$

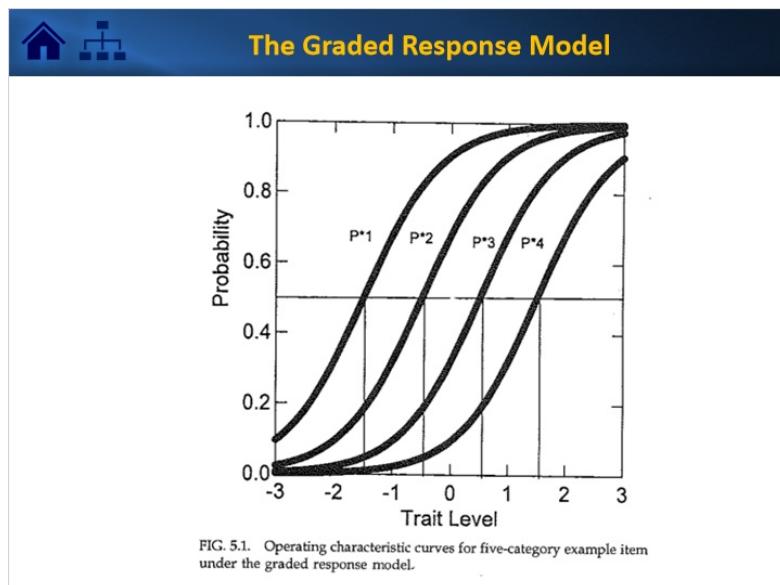
**Graded Response Model**

- Probability of a person's raw item score  $x$  falling in or above a given category  $k = 1, 2, \dots, m$  conditional on theta
- Operating characteristic curve

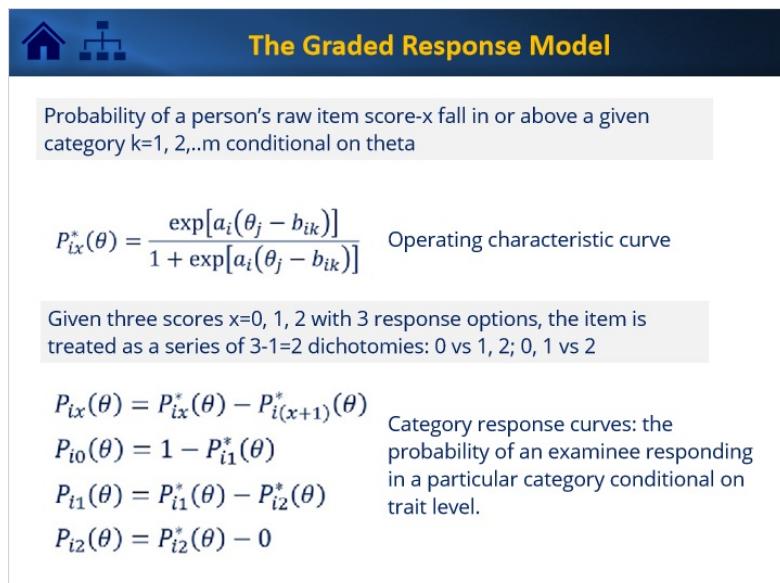
$$p_{ix}^*(\theta) = \frac{\exp[a_i(\theta_j - b_{ik})]}{1 + \exp[a_i(\theta_j - b_{ik})]}$$

- Category threshold represents the trait level to respond at or above threshold  $k$  with 0.5 probability.

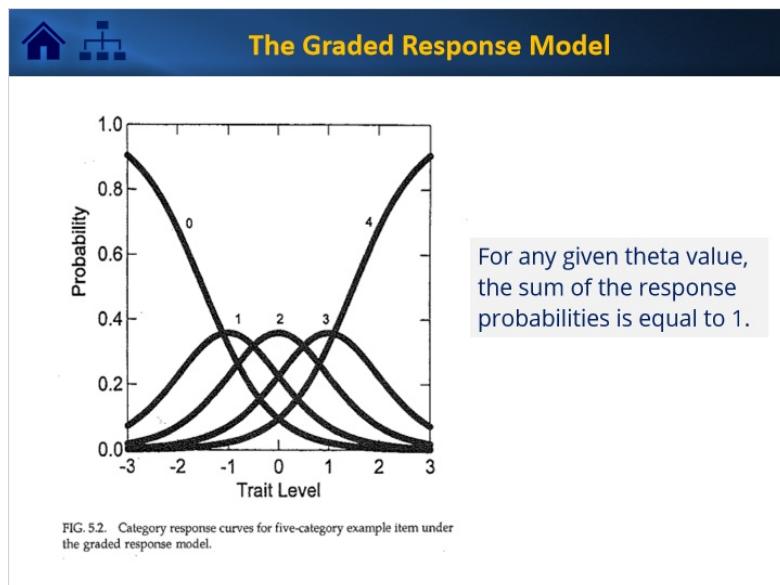
## 2.16 The Graded Response Model II



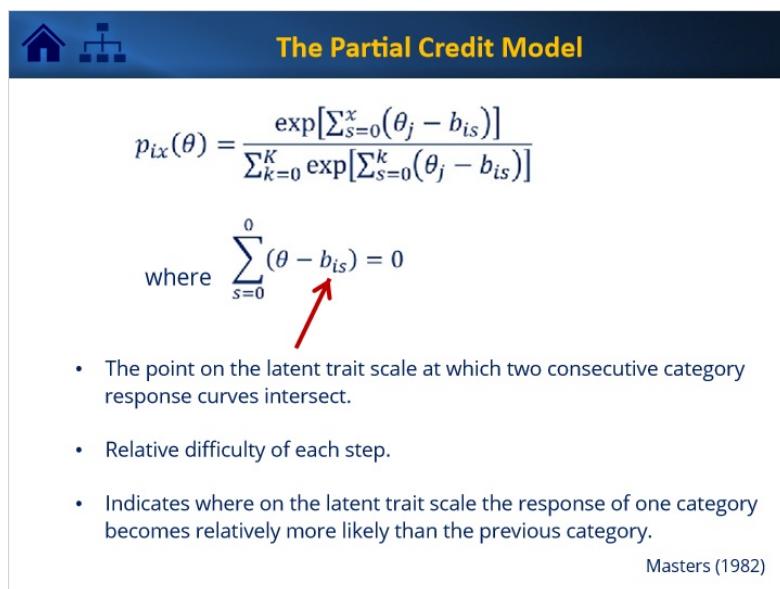
## 2.17 The Graded Response Model III



## 2.18 The Graded Response Model IV



## 2.19 The Partial Credit Model



## 2.20 The Reparameterized Partial Credit Model I

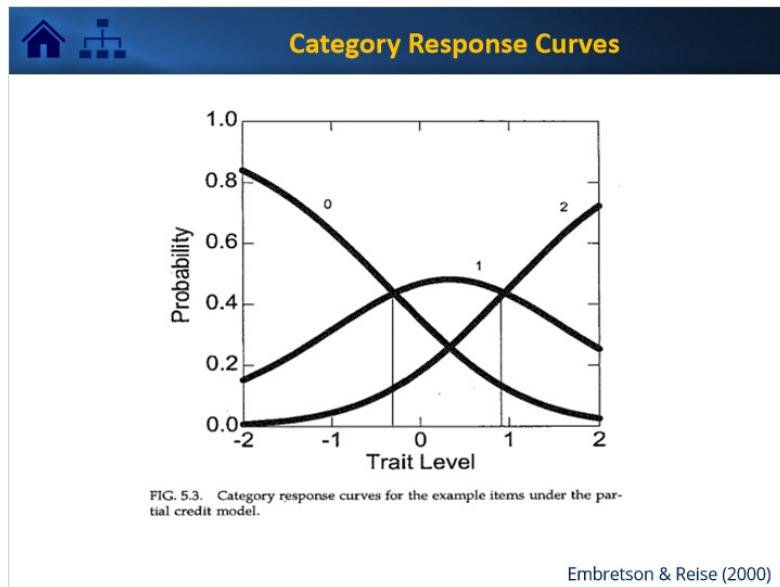
### The Reparameterized Partial Credit Model

Assume three score categories: 0, 1, 2

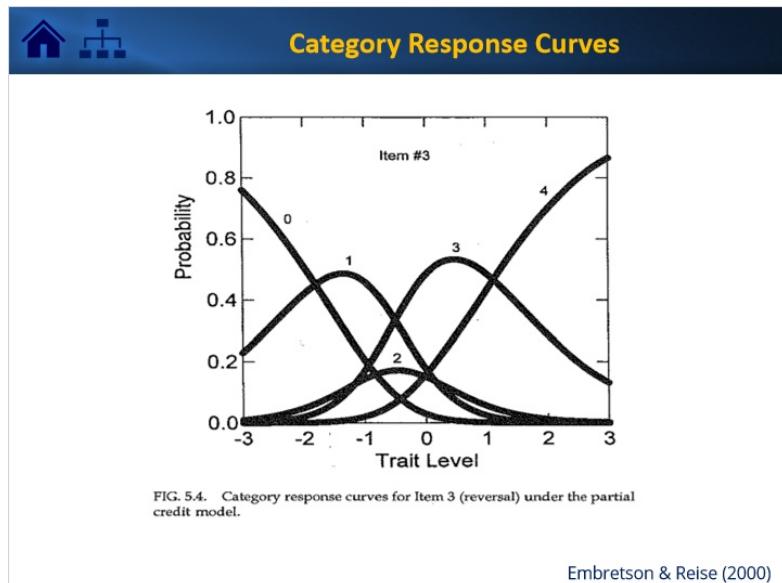
The probability of getting a certain score is

$$t_1 = \exp(\theta_j - b_i - d_{i1}) \quad t_2 = \exp(2\theta_j - 2b_i - d_{i1} - d_{i2})$$
$$p(x_{ji} = 0) = \frac{1}{1 + t_1 + t_2}$$
$$p(x_{ji} = 1) = \frac{t_1}{1 + t_1 + t_2}$$
$$p(x_{ji} = 2) = \frac{t_2}{1 + t_1 + t_2}$$

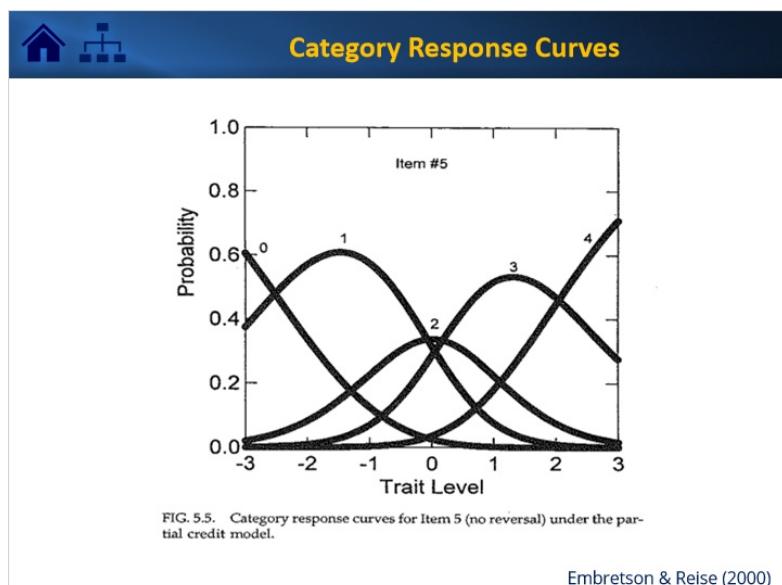
## 2.21 The Reparameterized Partial Credit Model II



## 2.22 The Reparameterized Partial Credit Model III



## 2.23 The Reparameterized Partial Credit Model IV



## 2.24 The Rating Scale Model

### The Rating Scale Model

$$p_{ix}(\theta) = \frac{\exp\left[\sum_{s=0}^x (\theta_j - (d_i + c_s))\right]}{\sum_{k=0}^K \exp\left[\sum_{s=0}^k (\theta_j - (d_i + c_s))\right]}$$

↑  
↑  
Category intersection parameter  
Item location parameter

$$\text{where } \sum_{s=0}^0 (\theta - (d_i + c_s)) = 0$$

Andrich (1978a, 1978b)

## 2.25 The Generalized Partial Credit Model I

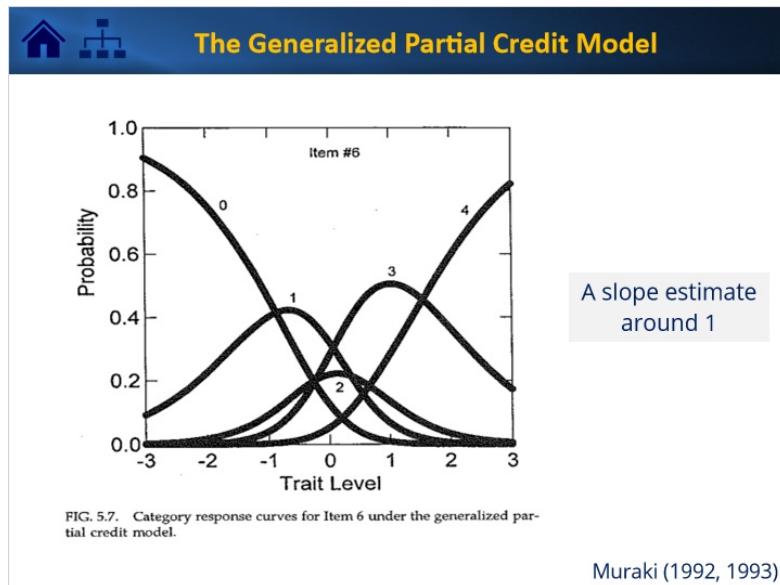
### The Generalized Partial Credit Model

$$p_{ix}(\theta) = \frac{\exp\left[\sum_{s=0}^x a_i(\theta_j - b_{is})\right]}{\sum_{k=0}^K \exp\left[\sum_{s=0}^k a_i(\theta_j - b_{is})\right]}$$
$$\text{where } \sum_{s=0}^0 a_i(\theta - b_{is}) = 0$$

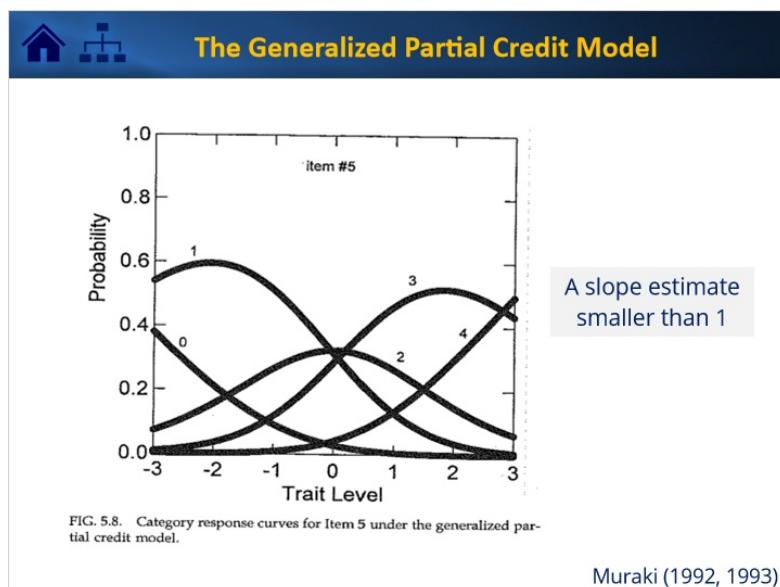
- The point on the latent trait scale at which two consecutive category response curves intersect.
- Indicates where on the latent trait scale the response of one category becomes relatively more likely than the previous category.
- The slope parameter indicate the degree to which categorical responses vary among items as theta level changes.

Muraki (1992, 1993)

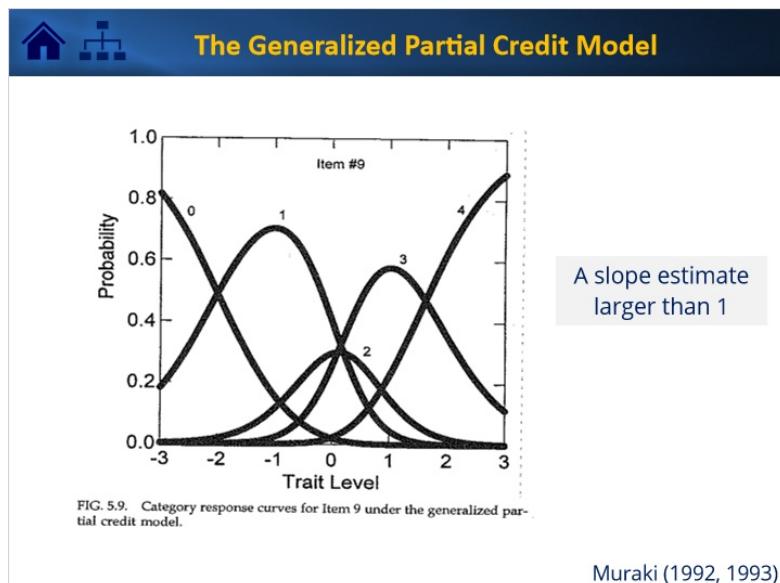
## 2.26 The Generalized Partial Credit Model II



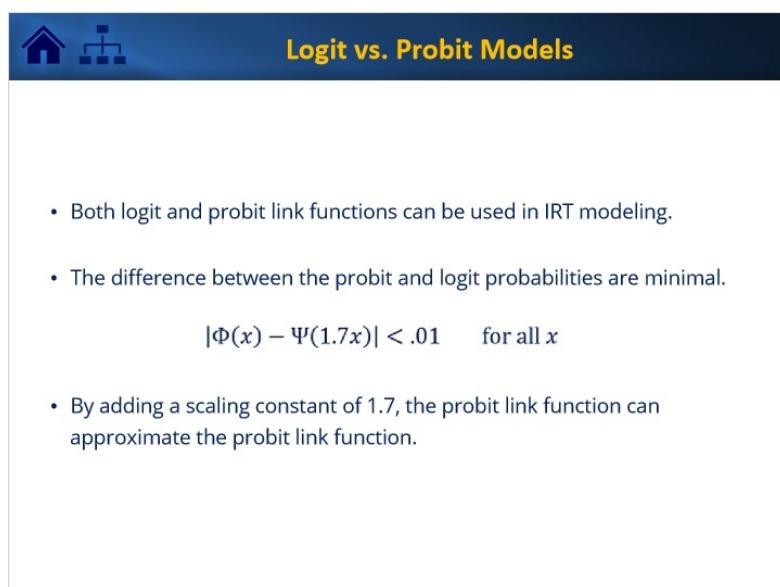
## 2.27 The Generalized Partial Credit Model III



## 2.28 The Generalized Partial Credit Model IV



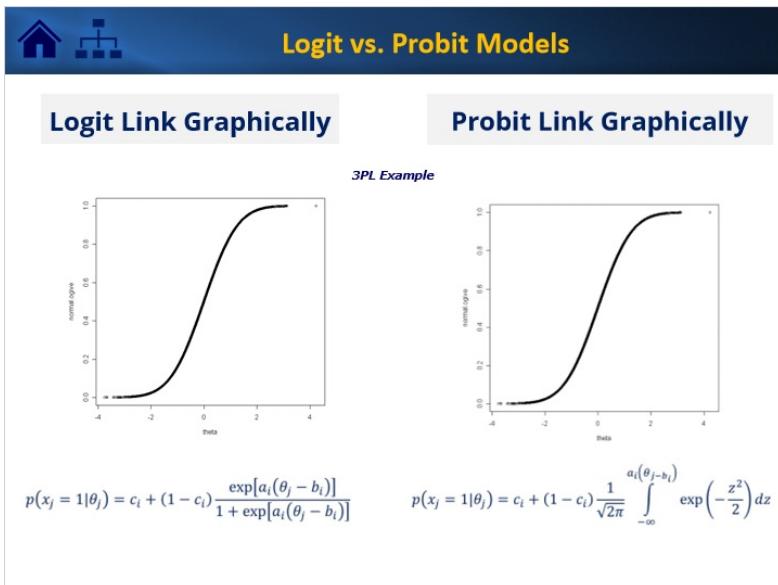
## 2.29 Logit vs. Probit Models I



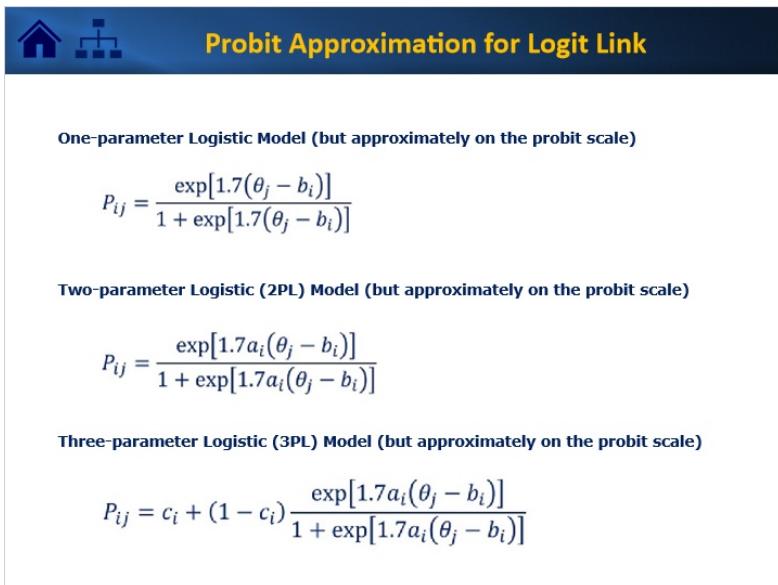
## 2.30 Logit vs. Probit Models II

Logit vs. Probit Models	
Logit Link	Probit Link
<b>The Rasch Measurement Model</b>	<b>One-parameter Probit Model</b>
$P_{ij} = \frac{\exp(\theta_j - b_i)}{1 + \exp(\theta_j - b_i)}$	$P_{ij} = \Phi(\theta_j - b_i)$
<b>Two-parameter Logistic (2PL) Model</b>	<b>Two-parameter Probit (2PP) Model</b>
$P_{ij} = \frac{\exp[a_i(\theta_j - b_i)]}{1 + \exp[a_i(\theta_j - b_i)]}$	$P_{ij} = \Phi[a_i(\theta_j - b_i)]$
<b>Three-parameter Logistic (3PL) Model</b>	<b>Three-parameter Probit (3PP) Model</b>
$P_{ij} = c_i + (1 - c_i) \frac{\exp[a_i(\theta_j - b_i)]}{1 + \exp[a_i(\theta_j - b_i)]}$	$P_{ij} = c_i + (1 - c_i)\Phi[a_i(\theta_j - b_i)]$

## 2.31 Logit vs. Probit Models III



## 2.32 Probit Approximation for Logit Link



The slide has a dark blue header bar. On the left is a blue house icon, followed by a blue icon of three squares connected by lines. To the right of these icons is the title "Probit Approximation for Logit Link" in yellow text.

**One-parameter Logistic Model (but approximately on the probit scale)**

$$P_{ij} = \frac{\exp[1.7(\theta_j - b_i)]}{1 + \exp[1.7(\theta_j - b_i)]}$$

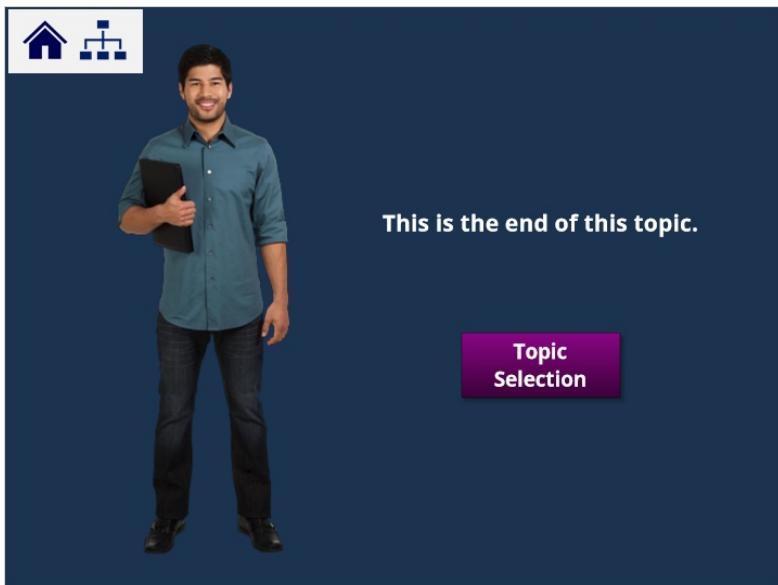
**Two-parameter Logistic (2PL) Model (but approximately on the probit scale)**

$$P_{ij} = \frac{\exp[1.7a_i(\theta_j - b_i)]}{1 + \exp[1.7a_i(\theta_j - b_i)]}$$

**Three-parameter Logistic (3PL) Model (but approximately on the probit scale)**

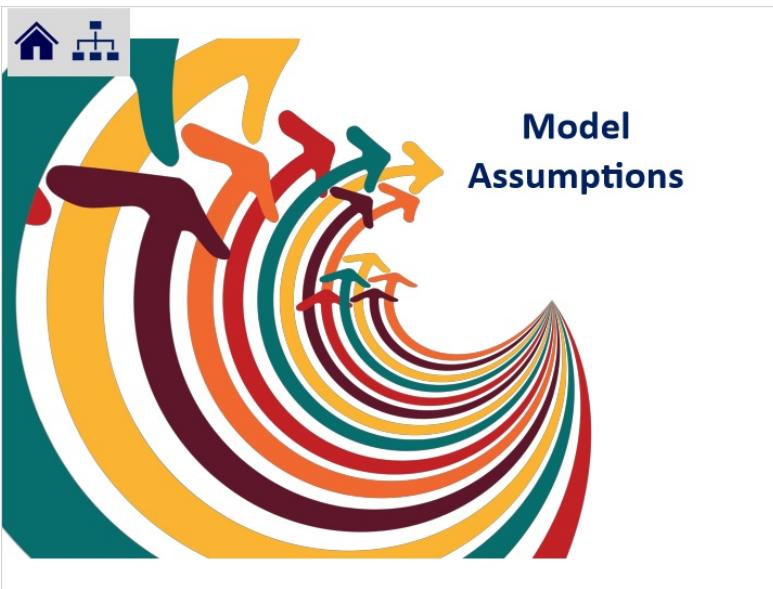
$$P_{ij} = c_i + (1 - c_i) \frac{\exp[1.7a_i(\theta_j - b_i)]}{1 + \exp[1.7a_i(\theta_j - b_i)]}$$

## 2.33 Bookend: Polytomous Models



The slide features a dark blue background. In the top left corner is a blue house icon, followed by a blue icon of three squares connected by lines. In the center-left, there is a full-body photograph of a smiling man in a teal button-down shirt and dark trousers, holding a black tablet or book. To the right of the man, the text "This is the end of this topic." is displayed in white. In the bottom right corner, there is a purple rectangular button with the white text "Topic Selection".

## 2.34 Bookmark: Model Assumptions



## 2.35 Assumptions for IRT Models

The slide has a dark blue header bar with a house icon and the title "Assumptions for IRT Models" in yellow. The main content area is a white box with a black border, containing the following list of assumptions:

- Unidimensionality
- Local independence
  - ✓ Local item independence-no item clustering
  - ✓ Local person independence-no person clustering
- Mathematical function for different models
  - ✓ Equal discrimination
  - ✓ No guessing
  - ✓ No slipping
- Non-speeded test
- One latent population

## 2.36 Local Independence



### Local Independence

**Local item independence**

$$p(U = u_y | \theta_j) = \prod_{i=1}^I p(u_i | \theta_j) = p(u_1 | \theta_j) p(u_2 | \theta_j) \dots p(u_I | \theta_j)$$

**Local person independence**

$$p(U_i = u_{iy} | \theta_j) = \prod_{j=1}^n p(u_j | \theta_j) = p(u_{i1} | \theta_1) p(u_{i2} | \theta_2) \dots p(u_{in} | \theta_n)$$

**Local independence**

$$p(U = u_y | \theta_j) = \prod_{j=1}^n \prod_{i=1}^I p(u_i | \theta_j)$$

(Embretson & Reise, 2000; Reckase, 2009)

## 2.37 Violations of Local Independence



### Violations of Local Independence

- **Violations of local item independence**
  - Item clustering due to manifest item grouping variables
    - ✓ Passage dependence
    - ✓ Item chaining
  - Latent item grouping variables
    - ✓ Latent item grouping measuring different dimensions of a latent construct
- **Violations of local person independence**
  - Person clustering due to observed grouping variables
    - ✓ Cluster sampling
    - ✓ Stratified sampling
  - Person clustering due to unknown latent grouping variables
    - ✓ Different problem-solving strategies
    - ✓ Latent differential item functioning

This ITEMS focuses on local item dependence!

## 2.38 Causes of Local Item Dependence



### Causes of Local Item Dependence

- **Passage dependence**
  - ✓ Passage-based reading comprehension test
  - ✓ Scenario-based science test
  - ✓ Table/graph-based math test
- **Item chaining (intentional)**
  - ✓ Multi-part items with order dependency
- **Explanation of previous answers (unintentional)**
  - ✓ Item clueing
- **Item or response format**
  - ✓ Multiple-choice vs constructed-response items
- **Content, knowledge, abilities**
  - ✓ Content clustering in science tests

(Hoskens & De Boeck, 1997; Yen, 1984)

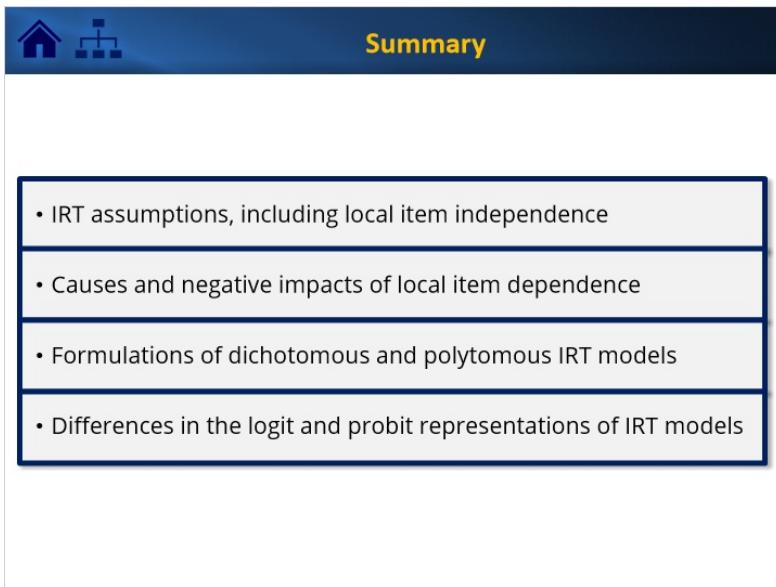
## 2.39 Bookend: Model Assumptions



This is the end of this topic.

**Topic Selection**

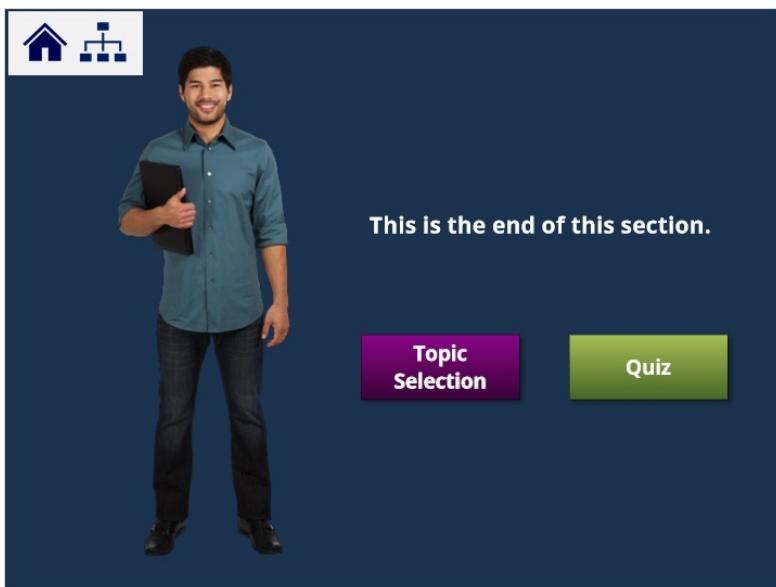
## **2.40 Summary**



The image shows a screenshot of a software interface titled "Summary". In the top left corner, there is a blue icon consisting of a house-like shape with three small squares above it. To the right of the icon, the word "Summary" is written in yellow capital letters. Below the title, there is a large white area containing a vertical list of four bullet points, each enclosed in a thin black border:

- IRT assumptions, including local item independence
- Causes and negative impacts of local item dependence
- Formulations of dichotomous and polytomous IRT models
- Differences in the logit and probit representations of IRT models

## **2.41 Bookend: Section 1**



The image shows a screenshot of a software interface featuring a male character standing on the left side. The character has dark hair and is wearing a teal button-down shirt and dark trousers, holding a black tablet or book. The background is a solid dark blue. In the center, the text "This is the end of this section." is displayed in white. At the bottom, there are two rectangular buttons: a purple one on the left labeled "Topic Selection" and a green one on the right labeled "Quiz".

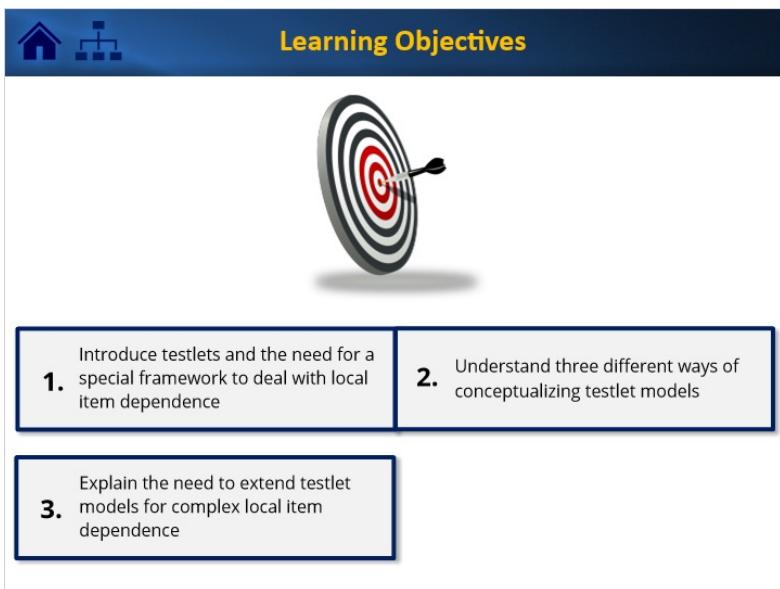
### 3. Section 2: Testlet Models

#### 3.1 Cover: Section 3



The image shows a classroom setup with rows of desks and chairs. A whiteboard at the front of the room displays the text "Hello" followed by "%LearnerName%". To the left of the whiteboard is a blue icon representing a house or home. To the right of the whiteboard, there is a large blue vertical banner with the text "Section 2: Testlet Models" in yellow and "[20 Minutes]" in white.

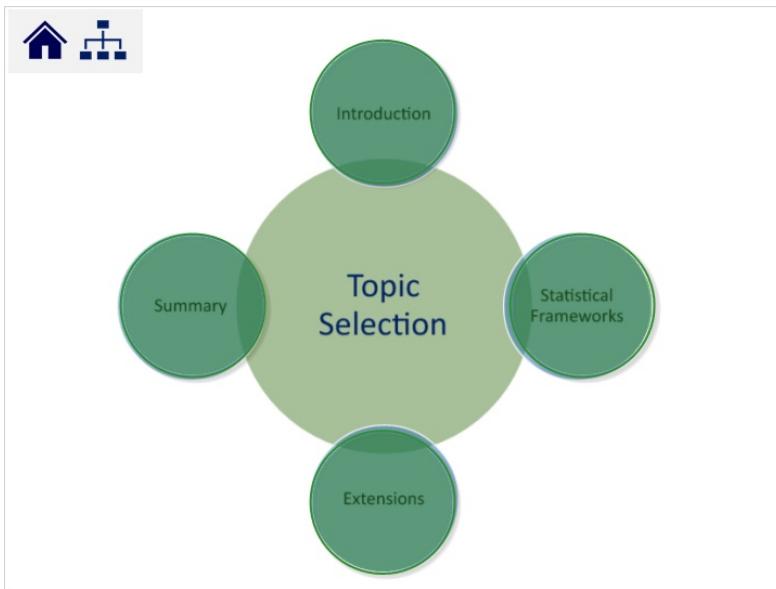
#### 3.2 Learning Objectives: Section 3



The slide has a dark blue header bar with a blue house icon and the text "Learning Objectives". Below the header is a large image of a target with an arrow hitting the bullseye. The slide is divided into three main sections, each containing a numbered objective:

- 1. Introduce testlets and the need for a special framework to deal with local item dependence
- 2. Understand three different ways of conceptualizing testlet models
- 3. Explain the need to extend testlet models for complex local item dependence

### 3.3 Topic Selection

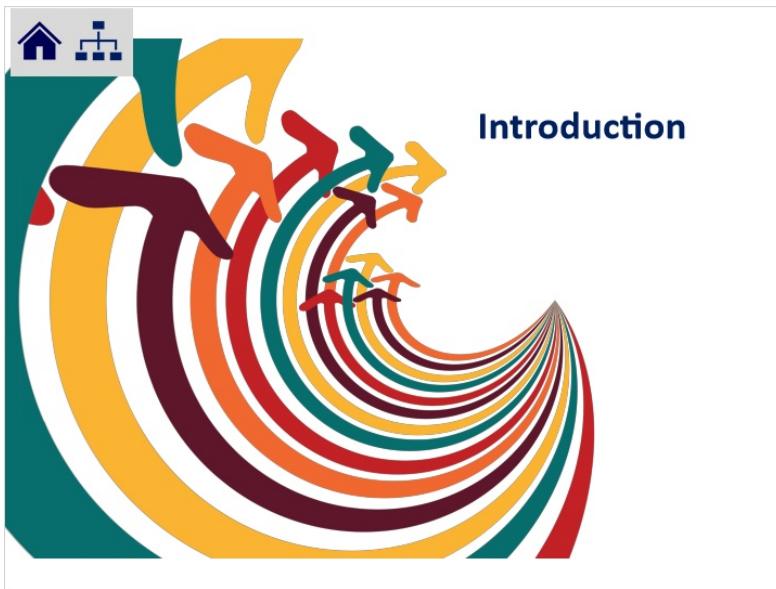


### References (Slide Layer)

#### References

- Hoskens and De Boeck (1997) and Tuerlinckx and De Boeck (1999) modeled item main effects and item interaction effects to account for LID: a constant Interaction model and a dimension-dependent Interaction model.
- Ip (2002) set up the reproducible and nonreproducible local dependence kernels to model LID based on conditional distributions describing multiple item responses as a function of ability without assuming local independence.
- Wang, Cheng, and Wilson (2005) used a multidimensional item response model to detect specific forms of LID for items across tests connected by common stimuli.
- Bayesian random-effects testlet models (Bradlow, Wainer, & Wang, 1999; Du, 1998; Wainer & Wang, 2000; Wang, Bradlow, & Wainer, 2002) developed to incorporate a parameter into unidimensional item response models, which indicates the interaction between person and item cluster.
- Rasch testlet model (Wang & Wilson, 2005) as a special case of multidimensional random coefficients multinomial logit model.
- Multilevel one-parameter testlet model (Jiao, Wang, & Kamata, 2005) in the framework of hierarchical generalized linear model.

### **3.4 Bookmark: Introduction**



### **3.5 What Is A Testlet**

**What Is A Testlet?**

- A testlet or an item bundle refers to a common stimulus with a cluster of items constructed around it.
- Some examples include passages in reading comprehension tests, scenarios in science assessment, tables or graphs in math assessment.
- In innovative assessment such as game-based or simulation-based assessment, testlets are the building blocks for creating a situational context for an assessment purpose.
- Testlets are common test construction units in many large-scale assessment programs such as PISA.
- The use of testlets helps to create an authentic context or situation to assess higher-order thinking skills.
- The use of testlets in general enhances the validity evidence collected in the process of assessment.

### **3.6 Testlets and LID**

#### Testlets and Local Item Dependence (LID)

- The use of testlets may impose challenges in psychometric analysis
- Items associated with the same stimulus are connected by the common context
  - Item connection or clustering may affect an examinee's performance on those items due to the common contextual effects
  - Thus, local item dependence (LID) or testlet effects may be induced.
- Local item dependence may negatively impact the quality of the psychometric analysis results.

### **3.7 Potential Impact of Testlet Effects**

#### Potential Impact of Testlet Effects

**When local item dependence is present, it is likely to lead to:**

- Overestimation of item discrimination
- Underestimation of item difficulty
- Overestimation of reliability, which may lead to early termination of a computerized adaptive test (CAT) when measurement precision is used for terminating a CAT
- Negative impact on equating results

### 3.8 Approaches to Testlet Models



### Approaches to Testlet Models

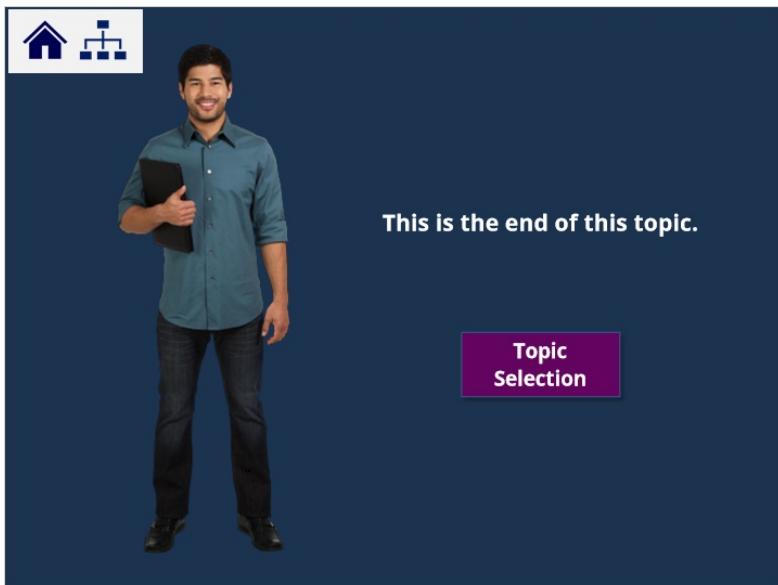
- **Random-effects modeling**
  - ✓ The Bayesian random-effects testlet model
- **Multidimensional modeling**
  - ✓ The Rasch testlet model
  - ✓ The generalized testlet model-bifactor structure
- **Multilevel modeling**
  - ✓ Hierarchical testlet model

### References (Slide Layer)

### References

- Hoskens and De Boeck (1997) and Tuerlinckx and De Boeck (1999) modeled item main effects and item interaction effects to account for LID: a constant Interaction model and a dimension-dependent Interaction model.
- Ip (2002) set up the reproducible and nonreproducible local dependence kernels to model LID based on conditional distributions describing multiple item responses as a function of ability without assuming local independence.
- Wang, Cheng, and Wilson (2005) used a multidimensional item response model to detect specific forms of LID for items across tests connected by common stimuli.
- Bayesian random-effects testlet models (Bradlow, Wainer, & Wang, 1999; Du, 1998; Wainer & Wang, 2000; Wang, Bradlow, & Wainer, 2002) developed to incorporate a parameter into unidimensional item response models, which indicates the interaction between person and item cluster.
- Rasch testlet model (Wang & Wilson, 2005) as a special case of multidimensional random coefficients multinomial logit model.
- Multilevel one-parameter testlet model (Jiao, Wang, & Kamata, 2005) in the framework of hierarchical generalized linear model.

### ***3.9 Bookend: Introduction***



### ***3.10 Bookmark: Statistical Frameworks***



### 3.11 The Random-Effects Testlet Model



#### The Random-Effects Testlet Model

- The two-parameter random-effects testlet model extends the 2PL IRT model by adding a random-effects parameter using probit link function (Bradlow, Wainer, & Wang, 1999)
$$P_{ij} = \Phi[a_i(\theta_j - b_i - \gamma_{jd(i)})]$$
- With logit link function:
$$p_{ij}(x_{ij} = 1 | \theta_j, \gamma_{jd(i)}, a_i, b_i) = \frac{1}{1 + \exp[-a_i(\theta_j - b_i - \gamma_{jd(i)})]}$$
- The testlet effect parameter,  $\gamma_{jd(i)}$ , is an interaction between person and a testlet
  - Constant for items associated with the same testlet
  - Magnitude of testlet effect quantified by its variance

### 3.12 The Two-Parameter Logistic Testlet Model



#### 2PL IRT vs 2PL Testlet Model

**2PL IRT**

$$P(Y_{ij} = 1) = \text{logit}^{-1}(t_{ij})$$
$$t_{ij} = a_i(\theta_j - b_i)$$

**2PL Testlet Model**

$$P(Y_{ij} = 1) = \text{logit}^{-1}(t_{ij})$$
$$t_{ij} = a_i(\theta_j - b_i - \boxed{\gamma_{jd(i)}})$$

### 3.13 The Rasch Testlet Model I

The Rasch Testlet Model

- Formulated from a multidimensional perspective
- Proposed as a special case of multidimensional random coefficients multinomial logit model by adding an additional dimension related to each testlet  
(Wang & Wilson, 2005)

$$p_{ij}(x_{ij} = 1 | \theta_j, \gamma_j, b_i) = \frac{1}{1 + \exp[-(\theta_j + \gamma_{jd(i)} - b_i)]}$$

- The ConQuest syntax on the next slide graphically shows the formulation of the multidimensional structure

### 3.14 The Rasch Testlet Model II

The Rasch Testlet Model

```
score (0,1) (0,1) (0,1) ( ) ( ) ! items(1);
score (0,1) (0,1) (0,1) ( ) ( ) ! items(2);
score (0,1) (0,1) ( ) (0,1) ( ) ! items(3);
score (0,1) (0,1) ( ) (0,1) ( ) ! items(4);
score (0,1) (0,1) ( ) ( ) (0,1) ! items(5);
score (0,1) (0,1) ( ) ( ) (0,1) ! items(6);
```

General Dimension    Testlet 1 Dimension    Testlet 2 Dimension    Testlet 3 Dimension

ConQuest syntax

### 3.15 The Generalized Testlet Model as a Bifactor Model



#### The Generalized Testlet Model as a Bifactor Model

- Testlet effects can be modeled as bifactor structure by treating each testlet effect as a secondary dimension

(Li, Bolt, & Fu, 2006; Rijmen, 2009)

$$p_{ij}(x_{ij} = 1 | \theta_j, \gamma_j, b_i) = \frac{1}{1 + \exp[-(a_i\theta_j + a_d\gamma_{jd(i)} - b_i)]}$$

- General factor represents the latent ability a test intends to measure while secondary factors are related to testlet effects
- Secondary factors account for within-testlet local item dependence (LID)

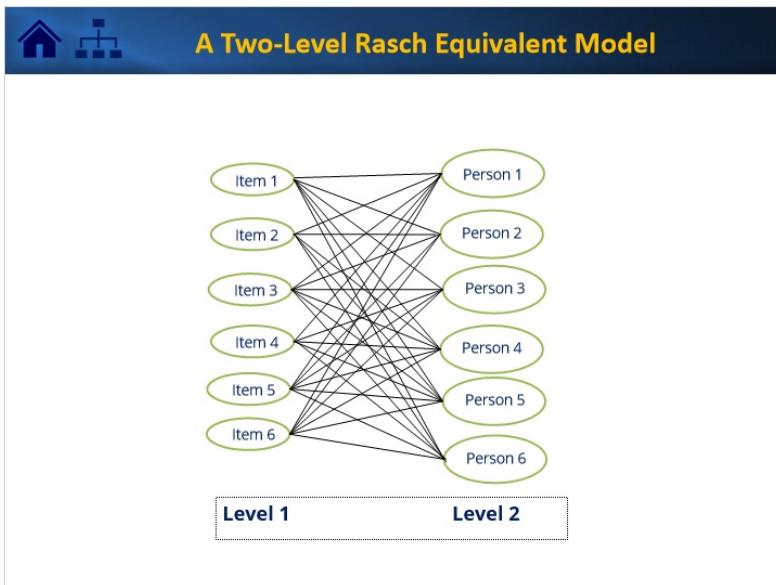
### 3.16 The Hierarchical IRT Model



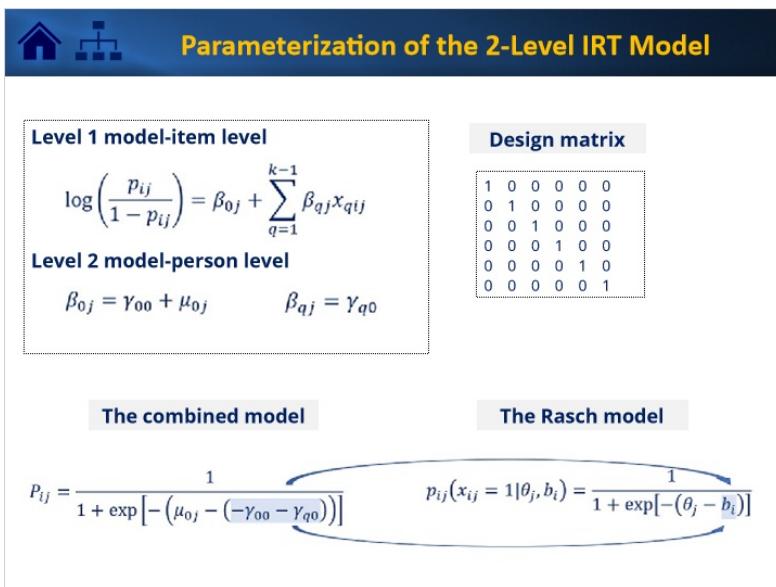
#### The Hierarchical IRT Model

- Reparameterization as hierarchical generalized linear model (HGLM) (Adams, Wilson, & Wu, 1997; Kamata, 1998, 2001)
- Reformulation of the Rasch model as a two-level HGLM
- Model parameters can be estimated using non-IRT software programs such as SAS, HLM, Mplus
- Descriptive IRT models can be expanded as explanatory IRT models by adding covariates
- Some psychometric issues such as DIF and explanations of DIF can be dealt with using the HGLM version of IRT models

### 3.17 A Two-Level Rasch Equivalent Model



### 3.18 Parameterization of the Two-Level IRT Model

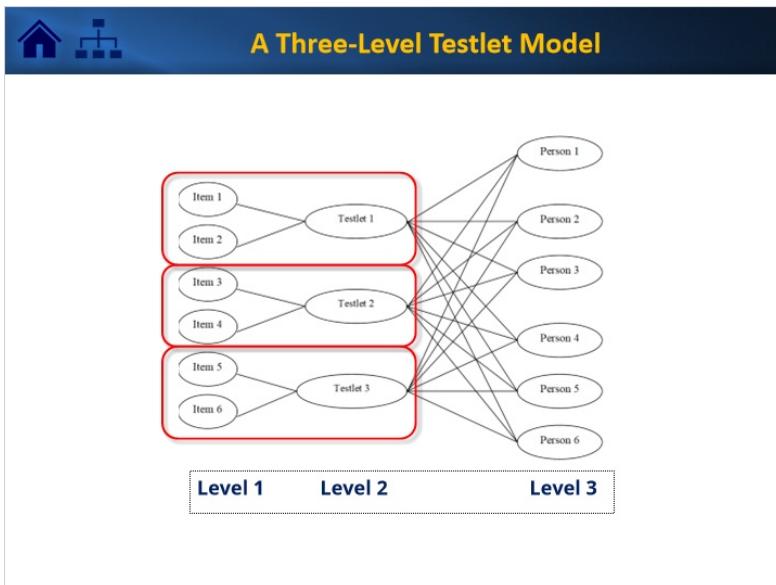


### 3.19 The Hierarchical Testlet Model

The Hierarchical Testlet Model

- Model testlet effects as item clustering in a hierarchical generalized linear model  
(Jiao, Wang, & Kamata, 2005)
- The reformulation of the Rasch testlet model as a three-level HGLM

### 3.20 A Three-Level Testlet Model



### 3.21 Parameterization of the Three-Level Testlet Model

  Parameterization of the 3-Level Testlet Model

**Level 1 model-item level**

$$\log\left(\frac{p_{idj}}{1 - p_{idj}}\right) = \beta_{0dj} + \sum_{q=1}^{k-1} \beta_{q dj} x_{qidj}$$

**Level 2 model-item group/testlet level**

$$\beta_{0dj} = \gamma_{00j} + \mu_{0dj} \quad \beta_{q dj} = \gamma_{q0j}$$

**Level 3 model-person level**

$$\gamma_{00j} = \pi_{000} + w_{00j} \quad \gamma_{q0j} = \pi_{q00}$$

$$P_{jdi} = \frac{1}{1 + \exp[-(w_{00j} - (-\pi_{000} - \pi_{q00}) + u_{0dj})]}$$

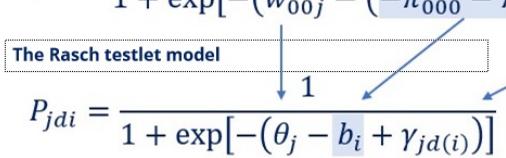
### 3.22 Equivalence of Rasch and Three-Level Testlet Models

  Equivalence of Rasch and 3-Level Testlet Models

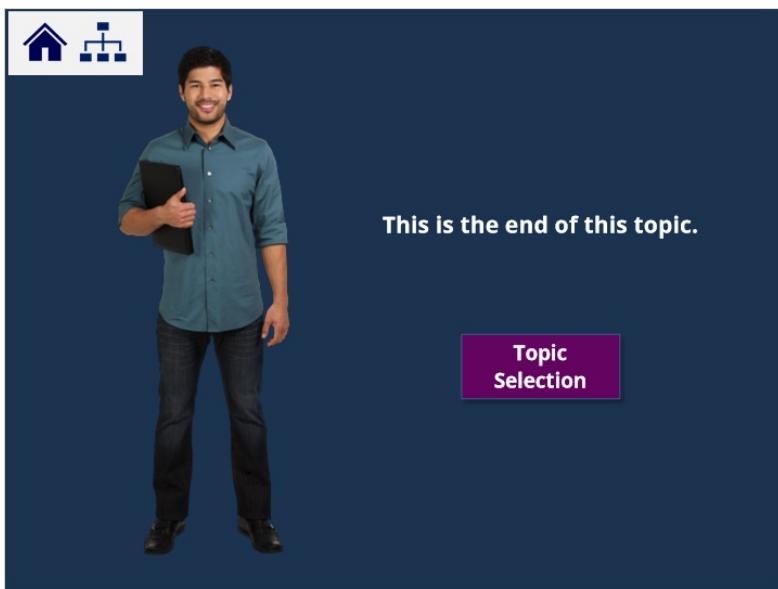
**The 3 levels combined model**

$$P_{jdi} = \frac{1}{1 + \exp[-(w_{00j} - (-\pi_{000} - \pi_{q00}) + u_{0dj})]}$$

**The Rasch testlet model**

$$P_{jdi} = \frac{1}{1 + \exp[-(\theta_j - b_i + \gamma_{jd(i)})]}$$


### **3.23 Bookend: Statistical Frameworks**



### **3.24 Bookmark: Extensions**



### **3.25 Testlet Models for More Complex Local Item Dependence Issues in Item Response Modeling**

**Testlet Models for More Complex LID Issues in IRT**

- Dual local item dependence due to cross-classified clustering structures from two grouping variables
- Directional local item dependence in multi-part items
- Between-testlet local item dependence in paired passages
  - ✓ Compensatory testlet model
  - ✓ Non-compensatory testlet model
- Simultaneous modeling of local item dependence and local person dependence

### **3.26 A Cross-Classified Testlet Model for Dual LID I**

**A Cross-Classified Testlet Model for Dual LID**

**Example: Science Assessment**

(Jiao, Wang, Wan, & Lu, 2009; Xie, 2014; Xie & Jiao, 2014)

```
graph TD; Content1((Content 1)) --- Item1_1((Item 1)); Content1 --- Item1_2((Item 1)); Content1 --- Item1_3((Item 1)); Content1 --- Item1_4((Item 1)); Content1 --- Item1_5((Item 1)); Content1 --- Item1_6((Item 1)); Content1 --- Item1_7((Item 1)); Content2((Content 2)) --- Item2_1((Item 2)); Content2 --- Item2_2((Item 2)); Content2 --- Item2_3((Item 2)); Content2 --- Item2_4((Item 2)); Content2 --- Item2_5((Item 2)); Content2 --- Item2_6((Item 2)); Content2 --- Item2_7((Item 2)); Item1_1 --- Target1((Target 1)); Item1_2 --- Target1; Item1_3 --- Target1; Item1_4 --- Target1; Item1_5 --- Target1; Item1_6 --- Target1; Item1_7 --- Target1; Item2_1 --- Target2((Target 2)); Item2_2 --- Target2; Item2_3 --- Target2; Item2_4 --- Target2; Item2_5 --- Target2; Item2_6 --- Target2; Item2_7 --- Target2; Target1 --- Person1_1((Person 1)); Target1 --- Person1_2((Person 1)); Target1 --- Person1_3((Person 1)); Target1 --- Person1_4((Person 1)); Target1 --- Person1_5((Person 1)); Target2 --- Person2_1((Person 2)); Target2 --- Person2_2((Person 2)); Target2 --- Person2_3((Person 2)); Target2 --- Person2_4((Person 2)); Target2 --- Person2_5((Person 2)); Level1[Level 1] --- Content1; Level1 --- Content2; Level2[Level 2] --- Target1; Level2 --- Target2; Level3[Level 3] --- Person1_1; Level3 --- Person1_2; Level3 --- Person1_3; Level3 --- Person1_4; Level3 --- Person1_5;
```

### 3.27 A Cross-Classified Testlet Model for Dual LID II



#### A Cross-Classified Testlet Model for Dual LID

- Items in a scenario-based science assessment might cluster due to the use of scenarios and content clustering such as physics, life science, earth science
- Items are cross-classified by scenarios and content domains
- Two testlet effect parameters,  $\gamma_{js(i)}$  and  $\gamma_{jc(i)}$ , are added into the model to account for each type of testlet effects due to scenario and content domain

$$p_{jdi} = \frac{1}{1 + \exp[-(\theta_j - b_i + \gamma_{js(i)} + \gamma_{jc(i)})]}$$

### 3.28 Testlet Models for More Complex LID Issues in IRT



#### Testlet Models for More Complex LID Issues in IRT

- Directional local item dependence in multi-part items
  - ✓ New item type developed for PARCC consortium tests, also called evidence-based selected response (EBSR)
  - ✓ Starts with a traditional selected-response item followed by a second selected-response item that asks students to show evidence from the text to support their answer to the first item
- A conditional IRT model for directional local item dependence based on item splitting (Marais & Andrich, 2008; Liao, Jiao, & Lissitz, 2016)
  - ✓ Proposed for context-dependent items (Baltruunas & Ricci 2009a & 2009b)
  - ✓ See also for response dependence (Marais & Andrich, 2008; Andrich & Kreiner, 2010; Andrich, Humphry, & Marais, 2012)

### 3.29 Item Splitting I




**Item Splitting**

Student	Item 1	Item 2 – Original	Item 2 – Recoded	
			Item 2a	Item 2b
1	1	1	1	x
2	0	1	x	1
3	0	0	x	0
4	0	1	x	1
5	1	0	0	x
6	1	1	1	x
7	1	1	1	x
8	0	0	x	0
9	1	0	0	x
10	0	1	x	1

### 3.30 Item Splitting II




**Item Splitting**

- IRT model for directional LID based on item splitting  
(Marais & Andrich, 2008; Liao, Jiao, & Lissitz, 2016)

$$P(x_{2j} = 1|x_{1j} = x) = \frac{\exp[a_2(\theta_j - (b_2 - xd))]}{1 + \exp[a_2(\theta_j - (b_2 - xd))]}$$

- Probability of second item conditional on getting first item wrong

$$P(x_{2j} = 1|x_{1j} = 0) = \frac{\exp[a_2(\theta_j - b_2)]}{1 + \exp[a_2(\theta_j - b_2)]}$$

- Probability of second item conditional on getting first item right  
(Includes non-negative parameter  $d$ )

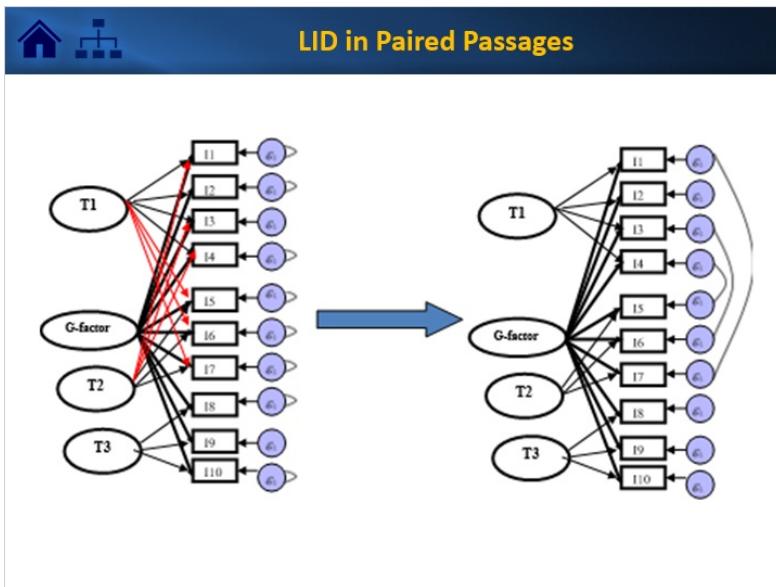
$$P(x_{2j} = 1|x_{1j} = 1) = \frac{\exp[a_2(\theta_j - (b_2 - d))]}{1 + \exp[a_2(\theta_j - (b_2 - d))]}$$

### 3.31 LID in Paired Passages I

LID in Paired Passages

- **Between-testlet LID in paired passages** which are intentionally made to be related to each other
- **Dual LID due to two paired passages** can be modeled using either
  - ✓ Compensatory structure, indicating that the higher level in one testlet ability may compensate for the deficiency in the other testlet ability
  - ✓ Non-compensatory structure, indicating that to answer an item correctly, an adequate level of proficiency on both testlet abilities is required to answer the item correctly

### 3.32 LID in Paired Passages II



### 3.33 Compensatory and Non-Compensatory Models



### Compensatory and Non-Compensatory Models

**A Non-Compensatory Two-Parameter Testlet Model**

$$P(x_{ij}|a_{1i}, a_{2i}, b_i, \theta_j, \gamma_{jt}) = \prod \frac{1}{1 + \exp[-(a_{1i} * \theta_j + a_{2i} * \gamma_{jt} - b_i)]}$$

**A Compensatory Two-Parameter Testlet Model (Jiao & Shu, 2014)**

$$P(x_{ij}|a_{1i}, a_{2i}, a_{3i}, b_i, \theta_j, \gamma_{jt1}, \gamma_{jt2}) = \frac{1}{1 + \exp[-(a_{1i} * \theta_j + a_{2i} * \gamma_{jt1} + a_{3i} * \gamma_{jt2} - b_i)]}$$

### 3.34 Dual LID and LPD



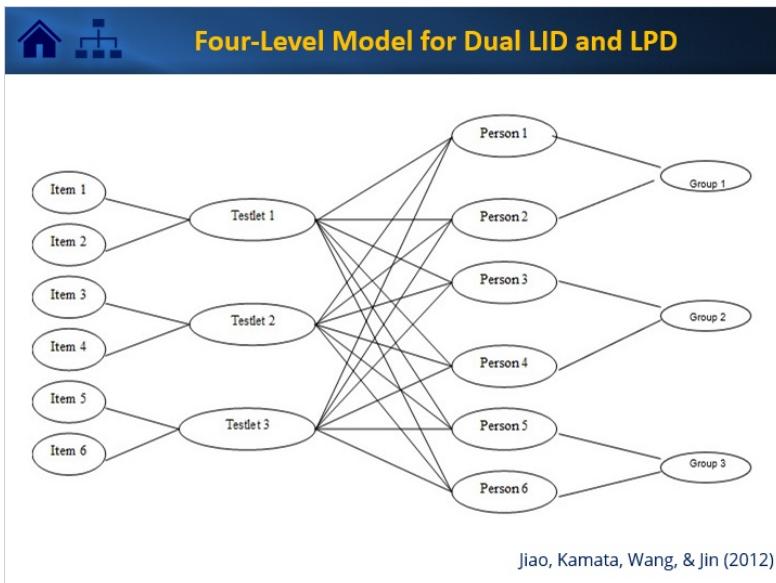
### Dual LID and LPD

Simultaneous modeling of local item dependence (LID) and local person dependence (LPD):

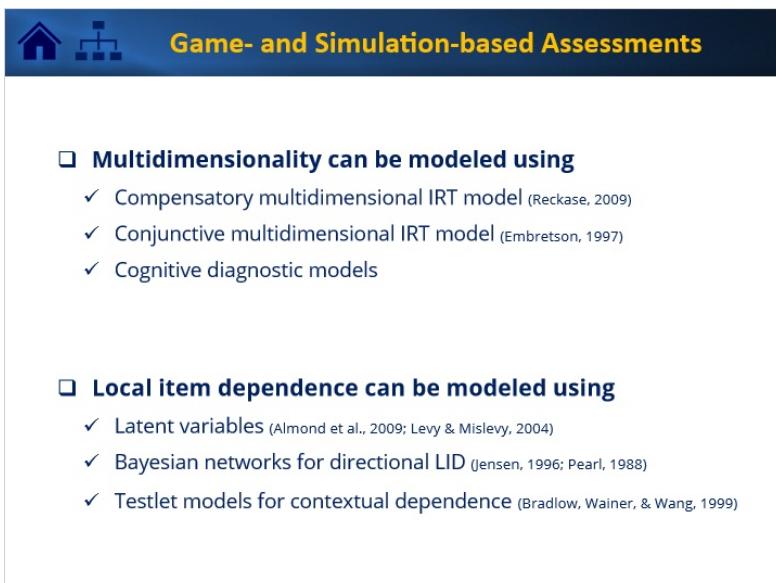
- Modeling of local item dependence and local person dependence simultaneously
- Person clustering effect is modeled by adding a group-specific ability parameter,  $\theta_g$
- Testlet effects are modeled as in the standard testlet model

$$P(x_{ij}|b_i, \theta_j, \gamma_{jt}, \theta_g) = \frac{1}{1 + \exp[-(\theta_j + \gamma_{jt(i)} + \theta_g - b_i)]}$$

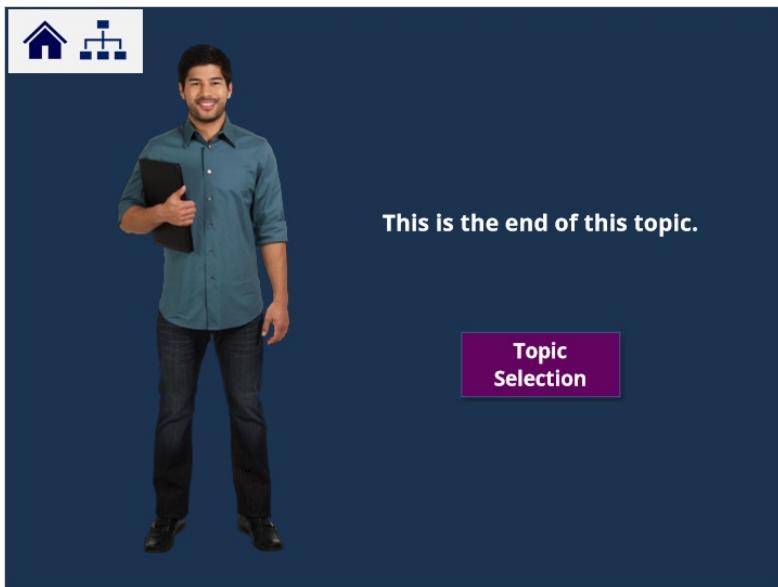
### 3.35 Four-Level Model for Dual LID and LPD



### 3.36 Cutting-Edge Examples



### **3.37 Bookend: Extensions**



### **3.38 Section Summary**

A slide titled "Section Summary". The header bar is blue and features a house icon and a three-tiered building icon on the left, and the title "Section Summary" in yellow text on the right. The main content area contains a bulleted list of topics related to testlet models and local item dependence (LID).

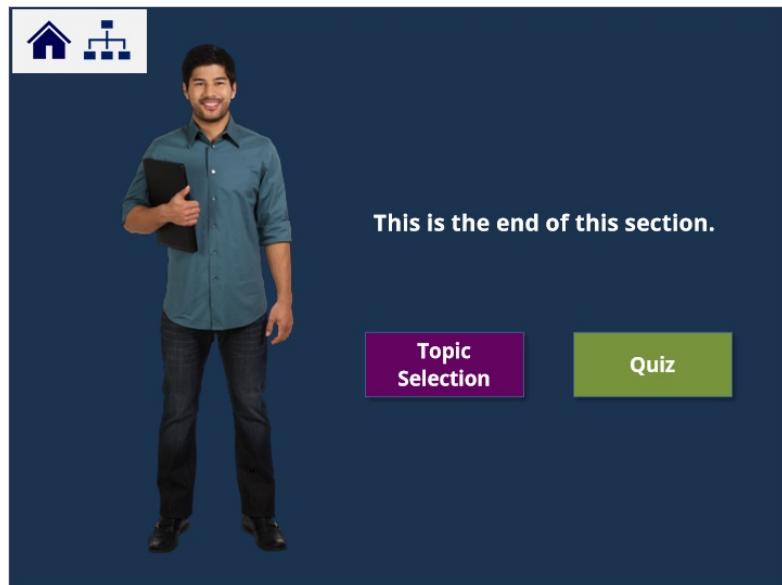
- Different formulation of testlet models
  - ✓ Random-effects modeling
  - ✓ Multidimensional perspective
  - ✓ Formulation as a hierarchical level in multilevel modeling
- Differences and similarities of the different perspectives to conceptualize the testlet effects
- More complex LID issues and the logic in developing extended models
  - ✓ Local item dependence due to two grouping variables leading to cross-classification structure
  - ✓ Direction local item dependence in multipart items
  - ✓ Dual local item dependence in items associated with paired passages
  - ✓ Concurrent modeling of local item and person dependence

## References (Slide Layer)

### References

- Hoskens and De Boeck (1997) and Tuerlinckx and De Boeck (1999) modeled item main effects and item interaction effects to account for LID: a constant Interaction model and a dimension-dependent Interaction model.
- Ip (2002) set up the reproducible and nonreproducible local dependence kernels to model LID based on conditional distributions describing multiple item responses as a function of ability without assuming local independence.
- Wang, Cheng, and Wilson (2005) used a multidimensional item response model to detect specific forms of LID for items across tests connected by common stimuli.
- Bayesian random-effects testlet models (Bradlow, Wainer, & Wang, 1999; Du, 1998; Wainer & Wang, 2000; Wang, Bradlow, & Wainer, 2002) developed to incorporate a parameter into unidimensional item response models, which indicates the interaction between person and item cluster.
- Rasch testlet model (Wang & Wilson, 2005) as a special case of multidimensional random coefficients multinomial logit model.
- Multilevel one-parameter testlet model (Jiao, Wang, & Kamata, 2005) in the framework of hierarchical generalized linear model.

### 3.39 Bookend: Section 3



## 4. Section 3: Estimation

### 4.1 Cover: Section 3



The slide features a photograph of a classroom with rows of desks and chairs. In the background, a whiteboard displays the text "Hello %LearnerName%". To the left of the photograph is a blue navigation bar with icons for home and sections. To the right of the photograph is a dark blue sidebar with yellow text.

**Section 3:**  
**Estimation**  
**[10 Minutes]**

### 4.2 Learning Objectives: Section 4



The slide features a target icon with an arrow hitting the bullseye, symbolizing achievement or success. Below the target are two numbered learning objectives in boxes.

**Learning Objectives**

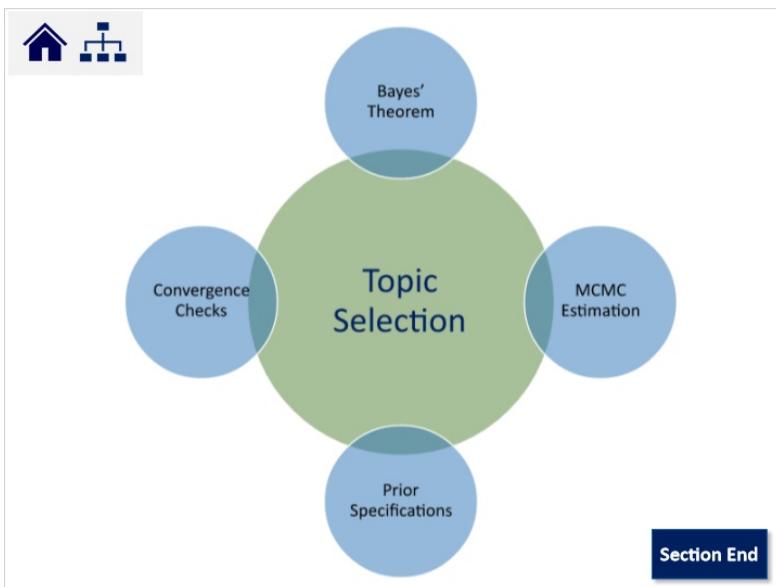
- 1.** Understand the basics of Bayesian estimation.
- 2.** Demonstrate how to estimate model parameters for the testlet models in OpenBUGS.

### 4.3 Summary

Overview

- Basics of Bayesian estimation
  - ✓ Bayes theorem
  - ✓ Specification of priors
  - ✓ Variable distributions
  - ✓ Convergence checks
- Estimation of model parameters for a testlet model, including both dichotomous and polytomous testlet models

### 4.4 Topic Selection



#### 4.5 Bookmark: Bayes' Theorem



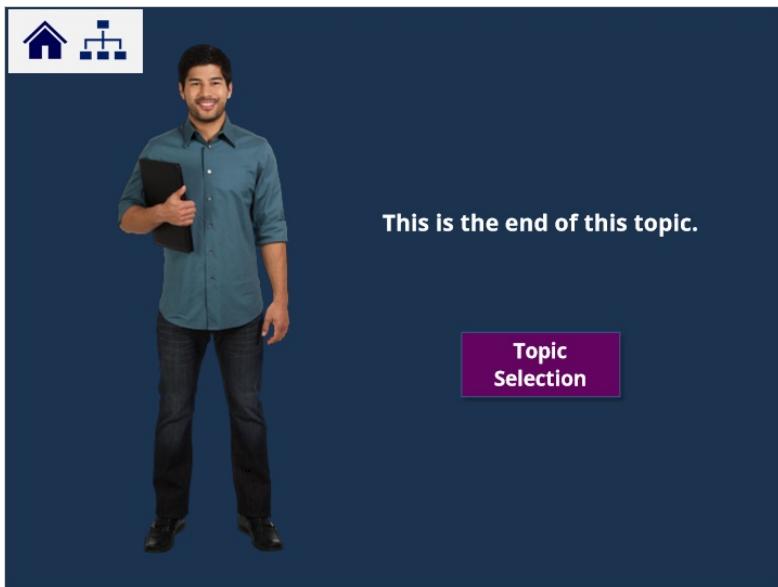
#### 4.6 Bayes' Theorem

$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$  The probability of A conditional on B

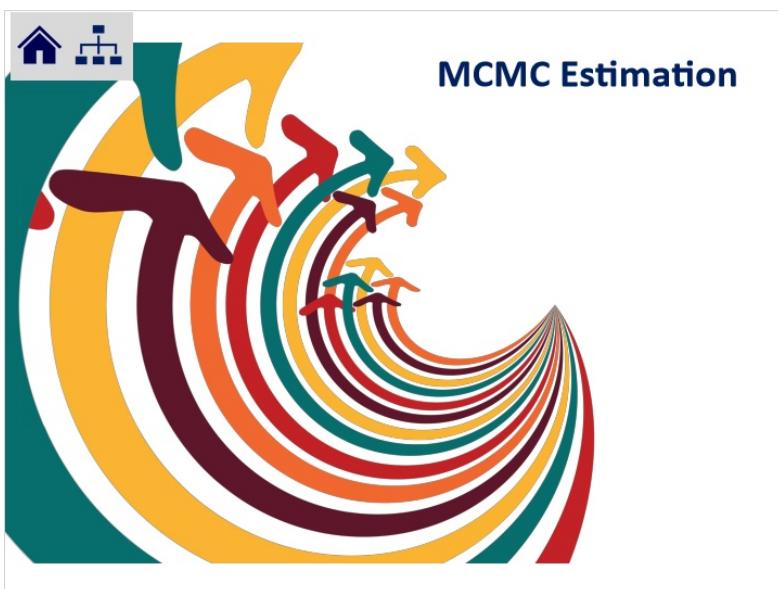
$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})}$  The probability of latent ability conditional on the observed item responses

Posterior =  $\frac{\text{(Conditional) Likelihood} \times \text{Prior}}{\text{Marginal Likelihood}} \propto \text{Likelihood} \times \text{Prior}$

#### **4.7 Bookend: Bayes' Theorem**



#### **4.8 Bookmark: MCMC Estimation**



## 4.9 Markov Chain Monte Carlo (MCMC) Estimation I

  **Markov Chain Monte Carlo (MCMC) Estimation**

- A **Markov chain** is a series of updated posterior distributions based on prior distributions and the likelihood (or data).
- **Model parameter estimates** are repeatedly sampled from their full conditional posterior distributions over a large number of iterations.
- A **sequence of  $W$**  (a vector of random variables) could be obtained as:  
$$\{W^0, W^1, W^2, \dots, W^t, W^{t+1}\}$$
- At each  $t + 1$  state,  $W^{t+1}$  is sampled from a **conditional distribution** of the current state,  $p(W^{t+1}|W^t)$ .
- $p(W^{t+1}|W^t)$  is the **transition kernel**  $k(W^t|W^{t+1})$ .
- The **transition kernel in IRT** is  
$$k[(\theta^t, \xi^t), (\theta^{t+1}, \xi^{t+1})] = p(\theta^{t+1}|\xi^t, Y)p(\xi^{t+1}|\theta^{t+1}, Y)$$

## 4.10 Markov Chain Monte Carlo (MCMC) Estimation II

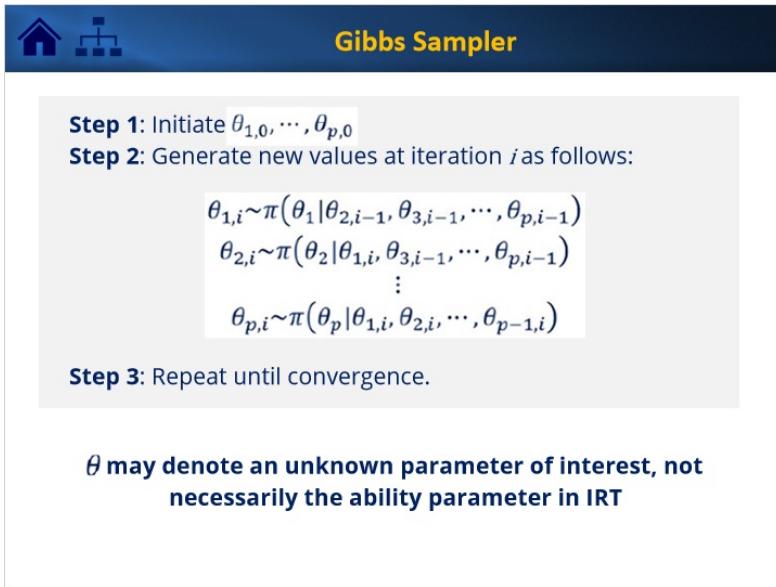
  **Markov Chain Monte Carlo (MCMC) Estimation**

Similar to **joint maximum likelihood estimation** method where item and person parameters are **estimated iteratively**

There is no need of integration over high-dimensional probability distributions 

Time intensive 

## 4.11 Gibbs Sampler



The slide has a blue header bar with a house icon and the text "Gibbs Sampler". The main content area is white with a grey sidebar on the left.

**Step 1:** Initiate  $\theta_{1,0}, \dots, \theta_{p,0}$

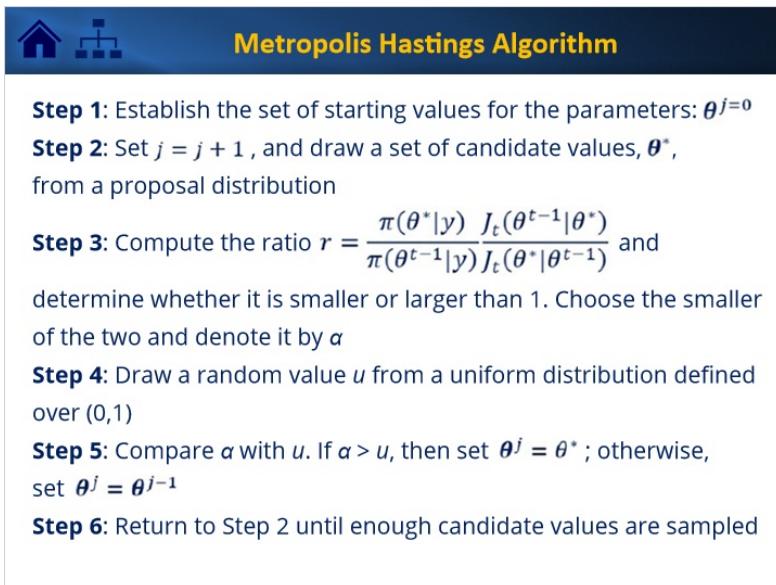
**Step 2:** Generate new values at iteration  $i$  as follows:

$$\begin{aligned}\theta_{1,i} &\sim \pi(\theta_1 | \theta_{2,i-1}, \theta_{3,i-1}, \dots, \theta_{p,i-1}) \\ \theta_{2,i} &\sim \pi(\theta_2 | \theta_{1,i}, \theta_{3,i-1}, \dots, \theta_{p,i-1}) \\ &\vdots \\ \theta_{p,i} &\sim \pi(\theta_p | \theta_{1,i}, \theta_{2,i}, \dots, \theta_{p-1,i})\end{aligned}$$

**Step 3:** Repeat until convergence.

**$\theta$  may denote an unknown parameter of interest, not necessarily the ability parameter in IRT**

## 4.12 Metropolis Hastings Algorithm



The slide has a blue header bar with a house icon and the text "Metropolis Hastings Algorithm". The main content area is white with a grey sidebar on the left.

**Step 1:** Establish the set of starting values for the parameters:  $\theta^{j=0}$

**Step 2:** Set  $j = j + 1$ , and draw a set of candidate values,  $\theta^*$ , from a proposal distribution

**Step 3:** Compute the ratio  $r = \frac{\pi(\theta^*|y) J_t(\theta^{t-1}|\theta^*)}{\pi(\theta^{t-1}|y) J_t(\theta^*|\theta^{t-1})}$  and determine whether it is smaller or larger than 1. Choose the smaller of the two and denote it by  $\alpha$

**Step 4:** Draw a random value  $u$  from a uniform distribution defined over (0,1)

**Step 5:** Compare  $\alpha$  with  $u$ . If  $\alpha > u$ , then set  $\theta^j = \theta^*$ ; otherwise, set  $\theta^j = \theta^{t-1}$

**Step 6:** Return to Step 2 until enough candidate values are sampled

## 4.13 Other Sampling Methods



### Other Sampling Methods

- **Adaptive rejection sampler** for log-concave conditional distribution for theta and item difficulty for item response theory models.
- **Slice sampling** for item discrimination parameters with a lognormal prior.
- All the sampling methods are **automatically selected** by OPENBUGS/WINBUGS.
- The sampling methods are held in Updater/Rsrc/Methods.odc **can be edited**:
  - ✓ For example, if there are problems with WinBUGS' adaptive rejection sampler (DFreeARS), then the method "UpdaterDFreeARS" for "log concave" could be replaced by "UpdaterSlice" (normally used for "real non linear")
  - ✓ This has been known to **sort out some Traps**. However, be careful and don't forget to keep a copy of the original Methods.odc file!

## 4.14 Bookend: MCMC Estimation



This is the end of this topic.

**Topic Selection**

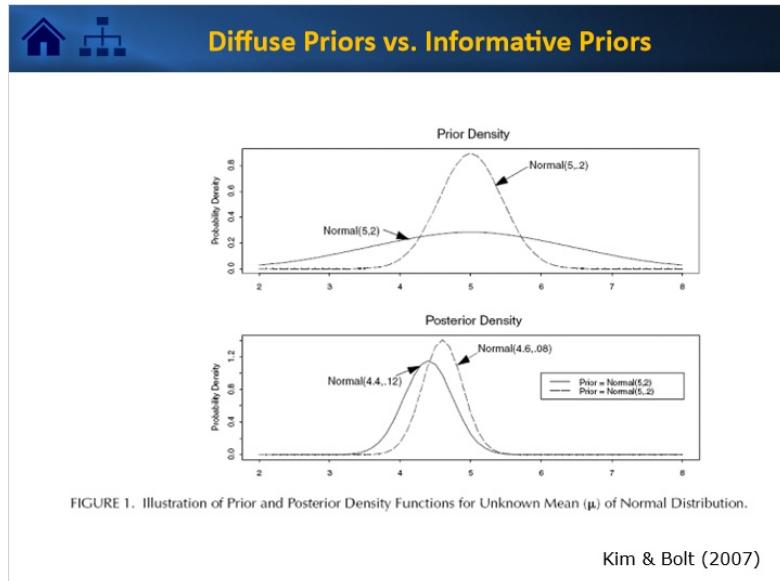
#### **4.15 Bookmark: Prior Specifications**



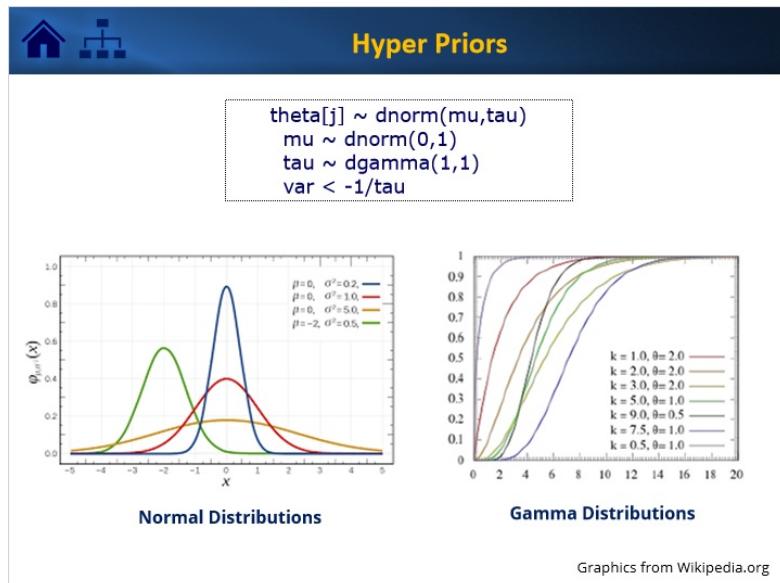
#### **4.16 Specification of Priors I**

The image shows a slide titled "Specification of Priors". The title is at the top left, followed by a blue decorative bar. The main content area contains a bulleted list of prior types: "Diffuse priors/flat priors", "Informative priors", "Hyper priors", and "Conjugate priors". To the right of the list is a cartoon illustration of a boy with brown hair, wearing a blue shirt and grey pants, pointing upwards with his right hand. A small lightbulb is shown above his head, symbolizing an idea or insight.

## 4.17 Specification of Priors II



## 4.18 Specification of Priors III



## 4.19 Specification of Priors IV

Prior $\pi(\theta)$	Data	Posterior $\pi(\theta y)$
$\pi(\theta) \equiv N(\mu, \tau^2)$	$Y_i \sim N(\theta, \sigma^2)$	$\pi(\theta y) \equiv N\left(\frac{\tau^2 n \bar{y} + \sigma^2 \mu}{n \tau^2 + \sigma^2}, \frac{\tau^2 \sigma^2}{n \tau^2 + \sigma^2}\right)$
$\pi(\theta) \equiv Beta(\alpha, \beta)$	$Y \sim Bin(n, \theta)$	$\pi(\theta y) \equiv Beta(\alpha + y, n + \beta - y)$
$\pi(\theta) \equiv Gamma(\alpha, \beta)$	$Y_i \sim Poisson(\theta)$	$\pi(\theta y) \equiv Gamma(\alpha + n \bar{y}, (n + \beta^{-1})^{-1})$
$\pi(\theta) \equiv InvGamma(\alpha, \beta)$	$Y_i \sim N(\mu, \theta)$	$\pi(\theta y) \equiv InvGamma(\alpha + \frac{n-1}{2}, (\frac{(n-1)s^2}{2} + \beta^{-1})^{-1})$

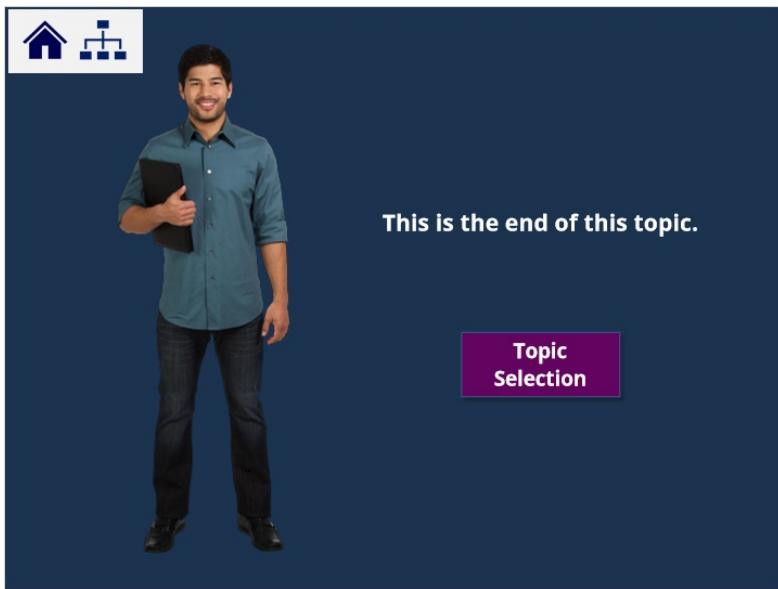
*Note.* In general, probability distributions that belong to an exponential family are the only ones with natural conjugate prior distributions and the posterior will involve the sufficient statistic for  $\theta$  (see Gelman, Carlin, Stern, & Rubin, 1995, p. 38, for details).

Rupp, Dey, & Zumbo (2004)

## 4.20 Specification of Priors V

Common Priors for IRT Models
<ul style="list-style-type: none"> <li>• <b>dnorm(<math>\mu, \tau</math>)</b> is the normal distribution or truncated normal distribution with parameters <math>\mu</math> and <math>\tau = 1/\sigma^2</math>, (0, 0.001), as flat prior: <math>\theta, b</math></li> <li>• <b>dnorm(<math>\mu, \tau</math>) I(L, U)</b> is the truncated normal distribution with parameters <math>\mu</math> and <math>\tau = 1/\sigma^2</math>: <math>a</math></li> <li>• <b>Lognorm (<math>\mu, \tau</math>)</b> is the log-normal distribution with parameters <math>\mu</math> and <math>\tau = 1/\sigma^2</math>, (0, 0.001), as flat prior: <math>a</math></li> <li>• <b>dbeta(a,b)</b> is the beta distribution with parameters <math>a</math> and <math>b</math>, (0.001, 0.001), as flat prior: <math>c, d</math></li> <li>• <b>dgamma(a,s)</b> is the gamma distribution, (0.001, 0.001), as flat prior: <math>\sigma^2</math></li> </ul>

#### **4.21 Bookend: Prior Specification**



#### **4.22 Bookmark: Convergence Checks**



## 4.23 Convergence Checks



### Convergence Check

Convergence should be checked based on **multiple criteria** to make sure that convergence was achieved before the model parameter estimates are monitored:

- Gelman-Rubin statistic (R) as modified by Brooks and Gelman (1998)
- Trace plots
- Quantile plots
- History plots
- Density plots

**Note on R**

## Details on R (Slide Layer)



### Supplementary Details on R

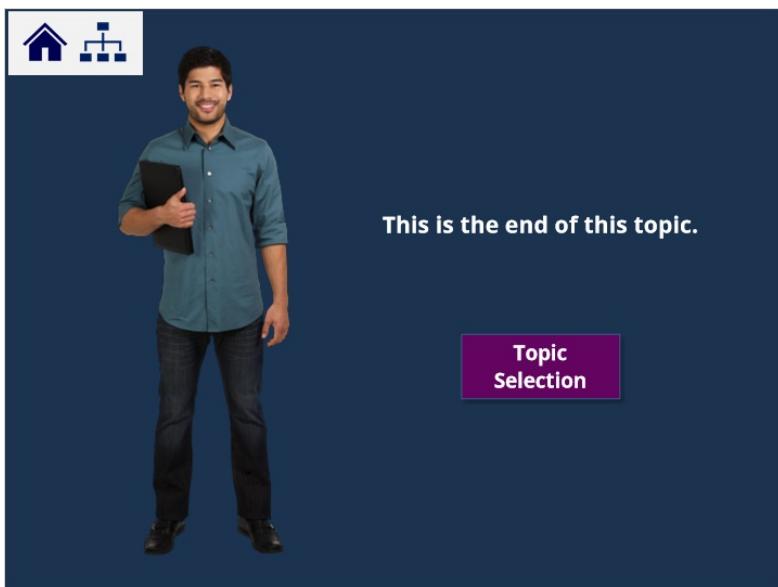
Convergence is assessed by comparing **within-chain (W)** and **between-chain (B) variability** over the second half of the chains.

The ratio  $R = B / W$  is expected to be **greater than 1** if the starting values were sufficiently different and can get **close to 1** as convergence is approached.

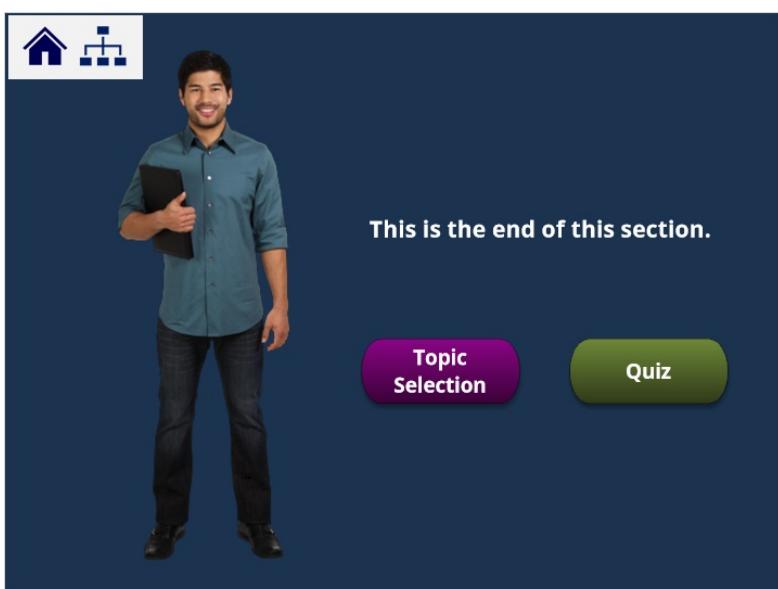
For practical purposes, **convergence can be assumed if  $R < 1.05$**  (Lunn et al., 2000). Brooks and Gelman (1998) emphasized the importance of ensuring not only that **R has converged to 1** but also that **B and W have converged to stability**.

Back

#### **4.24 Bookend: Convergence Checks**

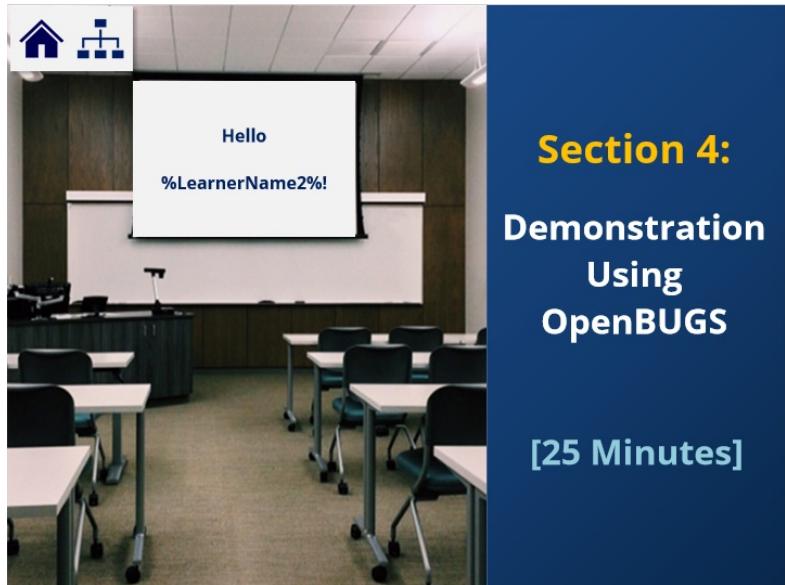


#### **4.25 Bookend: Section 4**

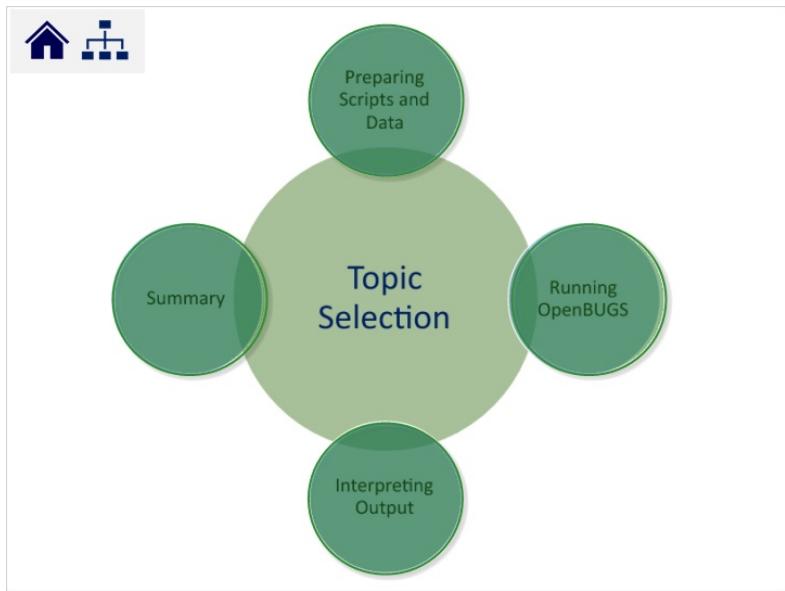


## 5. Section 4: Demonstration Using OpenBUGS

### 5.1 Cover: Section 5



### 5.2 Topic Selection



### 5.3 Bookmark: Preparing Script and Data



### 5.4 Model Specification I

The screenshot shows two windows of R code. The left window is titled 'testlet\_rasch' and contains the 'model' section of the script. It includes loops for data input, parameter estimation, and model identification. The right window is titled 'testlet\_rasch' and contains the 'specification of priors' section. It lists prior distributions for parameters like theta, gamma, and tau. A sidebar on the right provides a summary of the steps: 'Data to be used', 'The Rasch testlet model', 'Specification of priors', and 'Model identification'.

```
testlet_rasch
model
# data to be used
for (i in 1:2) {
  for (j in 1:10) {
    r[i,j] ~ dnorm(0,tau)
  }
}
# IRT model
for (i in 1:2) {
  for (j in 1:10) {
    p[i,j] <- (exp(theta[i][j]))/(1+exp(theta[i][j]+gamma[i][j]))
  }
}
for (i in 0:10) {
  p[0,i] <- (exp(theta[0][i])-gamma[0][i])/((1-exp(theta[0][i])-gamma[0][i])+1)
}
for (i in 11:15) {
  p[0,i] <- (exp(theta[0][i])-gamma[0][i])/((1-exp(theta[0][i])-gamma[0][i])+1)
}
for (i in 16:20) {
  p[0,i] <- (exp(theta[0][i])-gamma[0][i])/((1-exp(theta[0][i])-gamma[0][i])+1)
}
for (i in 21:25) {
  p[0,i] <- (exp(theta[0][i])-gamma[0][i])/((1-exp(theta[0][i])-gamma[0][i])+1)
}

# specification of priors
for (j in 1:10) {
  theta[j] ~ dnorm(mu_tau,tau)
  gamma[1,j] ~ dnorm(mu_gamma1,tau1)
  gamma[2,j] ~ dnorm(mu_gamma2,tau2)
  gamma[3,j] ~ dnorm(mu_gamma3,tau3)
  gamma[4,j] ~ dnorm(mu_gamma4,tau4)
  gamma[5,j] ~ dnorm(mu_gamma5,tau5)
}
mu_dnorm(0,1)
var_dnorm(1,1)
var_1tau ~ dtau(0,1)
tau1 ~ dgamma(1,1)
tau2 ~ dgamma(1,1)
tau3 ~ dgamma(1,1)
tau4 ~ dgamma(1,1)
tau5 ~ dgamma(1,1)
var_gamma1 ~ tmaut1
var_gamma2 ~ tmaut2
var_gamma3 ~ tmaut3
var_gamma4 ~ tmaut4
var_gamma5 ~ tmaut5
# model identification
theta[1,1] ~ dnorm(0,1)
tau1 ~ dtau(0,1)
tau[0,1] <- -1*sum(tau[1:5])
```

## 5.5 Model Specification III

**Model Specification**

```

testlet_rasch
model
# data to be used
for (i in 1:2) {
  for (j in 1:10) {
    r[i,j] ~ dbern(p[i,j])
  }
}

#IRT model
for (i in 1:2) {
  for (j in 1:10) {
    p[i,j] ~ dexp(theta[i]-gamma[1]+b[j])/(1+dexp(theta[i]-gamma[1]+b[j]))
  }
}
for (i in 6:10) {
  p[i,j] ~ dexp(theta[i]-gamma[2]+b[j])/(1+dexp(theta[i]-gamma[2]+b[j]))
}

}
for (i in 11:15) {
  p[i,j] ~ dexp(theta[i]-gamma[3]+b[j])/(1+dexp(theta[i]-gamma[3]+b[j]))
}

}
for (i in 16:20) {
  p[i,j] ~ dexp(theta[i]-gamma[4]+b[j])/(1+dexp(theta[i]-gamma[4]+b[j]))
}

}
for (i in 21:25) {
  p[i,j] ~ dexp(theta[i]-gamma[5]+b[j])/(1+dexp(theta[i]-gamma[5]+b[j]))
}
}

# specification of priors
for (i in 1:5) {
  theta[i] ~ dnorm(mu_theta)
  gamma[i] ~ dnorm(0,taut)
  gamma[6] ~ dnorm(0,taut2)
  gamma[7] ~ dnorm(0,taut2)
  gamma[8] ~ dnorm(0,taut4)
  gamma[9] ~ dnorm(0,taut4)
  gamma[10] ~ dnorm(0,taut5)
  mu_theta ~ dnorm(0,1)
  taut ~ dgamma(1,1)
  var_theta ~ dgamma(1,1)
  taut2 ~ dgamma(1,1)
  taut4 ~ dgamma(1,1)
  var_gamma[1] ~ taud1
  var_gamma[2] ~ taud2
  var_gamma[3] ~ taud3
  var_gamma[4] ~ taud4
  var_gamma[5] ~ taud5
}

# model identification
for (i in 1:10) {
  b[i] ~ dnorm(0,1)
}
b[6] <- -1*sum(b[1:5])
}

```

- Data to be used
- The Rasch testlet model
- Specification of priors
- Model identification

## 5.6 Model Specification III

**Model Specification**

```

testlet_rasch
model
# data to be used
for (i in 1:2) {
  for (j in 1:10) {
    r[i,j] ~ dbern(p[i,j])
  }
}

#IRT model
for (i in 1:2) {
  for (j in 1:10) {
    p[i,j] ~ dexp(theta[i]-gamma[1]+b[j])/(1+dexp(theta[i]-gamma[1]+b[j]))
  }
}
for (i in 6:10) {
  p[i,j] ~ dexp(theta[i]-gamma[2]+b[j])/(1+dexp(theta[i]-gamma[2]+b[j]))
}

}
for (i in 11:15) {
  p[i,j] ~ dexp(theta[i]-gamma[3]+b[j])/(1+dexp(theta[i]-gamma[3]+b[j]))
}

}
for (i in 16:20) {
  p[i,j] ~ dexp(theta[i]-gamma[4]+b[j])/(1+dexp(theta[i]-gamma[4]+b[j]))
}

}
for (i in 21:25) {
  p[i,j] ~ dexp(theta[i]-gamma[5]+b[j])/(1+dexp(theta[i]-gamma[5]+b[j]))
}
}

# specification of priors
for (i in 1:5) {
  theta[i] ~ dnorm(mu_theta)
  gamma[1] ~ dnorm(0,taut)
  gamma[2] ~ dnorm(0,taut2)
  gamma[3] ~ dnorm(0,taut)
  gamma[4] ~ dnorm(0,taut)
  gamma[5] ~ dnorm(0,taut)
  mu_theta ~ dnorm(0,1)
  taut ~ dgamma(1,1)
  var_theta ~ dgamma(1,1)
  taut2 ~ dgamma(1,1)
  taut ~ dgamma(1,1)
  taut ~ dgamma(1,1)
  var_gamma[1] ~ taud1
  var_gamma[2] ~ taud2
  var_gamma[3] ~ taud3
  var_gamma[4] ~ taud4
  var_gamma[5] ~ taud5
}

# model identification
for (i in 1:10) {
  b[i] ~ dnorm(0,1)
}
b[6] <- -1*sum(b[1:5])
}

```

- Data to be used
- The Rasch testlet model
- Specification of priors
- Model identification

## 5.7 Model Specification IV

**Model Specification**

```

model
  # data to be used
  for (i in 1:2) {
    for (j in 1:10) {
      r[i,j] ~ dnorm(0,tais)
    }
  }

  # IRT model
  for (i in 1:2) {
    for (j in 1:10) {
      p[i,j] <- exp(theta[i]-gamma[1][j])/(1+exp(theta[i]-gamma[1][j]))
    }
    for (i in 0:10) {
      p[i,j] <- exp(theta[i]-gamma[2][j])/(1+exp(theta[i]-gamma[2][j]))
    }
  }

  for (i in 11:15) {
    p[i,j] <- exp(theta[i]-gamma[3][j])/(1+exp(theta[i]-gamma[3][j]))
  }

  for (i in 16:20) {
    p[i,j] <- exp(theta[i]-gamma[4][j])/(1+exp(theta[i]-gamma[4][j]))
  }

  for (i in 21:25) {
    p[i,j] <- exp(theta[i]-gamma[5][j])/(1+exp(theta[i]-gamma[5][j]))
  }
}

specification of priors
for (i in 1:5) {
  theta[i] ~ dnorm(0,tais)
  gamma[1][i] ~ dnorm(0,taut1)
  gamma[2][i] ~ dnorm(0,taut2)
  gamma[3][i] ~ dnorm(0,taut3)
  gamma[4][i] ~ dnorm(0,taut4)
  gamma[5][i] ~ dnorm(0,taut5)

  tau[1,i] ~ dnorm(0,1)
  tau[2,i] ~ dnorm(0,1)
  var ~ r1tau
  taut1 ~ dnorm(0,1)
  taut2 ~ dnorm(0,1)
  taut3 ~ dnorm(0,1)
  taut4 ~ dnorm(0,1)
  taut5 ~ dnorm(0,1)
  var_gamma[1] ~ r1taud1
  var_gamma[2] ~ r1taud2
  var_gamma[3] ~ r1taud3
  var_gamma[4] ~ r1taud4
  var_gamma[5] ~ r1taud5
}

# model identification
theta[1] ~ dnorm(0,1)
b[1] <- -1*sum(b[1:10])

```

- Data to be used
- The Rasch testlet model
- Specification of priors
- Model identification

## 5.8 Model Specification V

**Model Specification**

In IRT models, the difference between latent trait and item difficulty predicts the log odds. For the same log odds, different combinations of person trait and item difficulty will make the same prediction. For example,

$$\ln\left(\frac{P_{ij}}{1 - P_{ij}}\right) = a_i(\theta_j - b_i)$$

$$1.5 = 1(3-1.5)$$

$$1.5 = 1(2-0.5)$$

$$1.5 = 1(1-(-0.5))$$

$$1.5 = 2(1-0.25)$$

Anchor on either the person or item side

## 5.9 Model Specification VI

**Model Specification**

```

testlet_rasch
model
{
  # data to be used
  for (i in 1:2) {
    for (l in 1:10) {
      r[i,l] ~ dbern(p[i,l])
    }
  }

  #IRT model
  for (i in 1:2) {
    for (l in 1:10) {
      p[i,(i-1)*10+l] ~ dexp(theta[i]+gamma[1]*b[i])
    }
  }
  for (i in 0:10) {
    p[0,(i-1)*10+l] ~ dexp(theta[i]+gamma[2]*b[i])
  }
}

for (i in 1:15) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[3]*b[i])
}
for (i in 16:20) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[4]*b[i])
}
for (i in 21:25) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[5]*b[i])
}

```

```

#specification of priors
for (i in 1:10) {
  theta[i] ~ dnorm(0,tau)
  gamma[1]~dnorm(0,tau1)
  gamma[2]~dnorm(0,tau2)
  gamma[3]~dnorm(0,tau3)
  gamma[4]~dnorm(0,tau4)
  gamma[5]~dnorm(0,tau5)
  tau ~ dnorm(0,1)
  tau1 ~ dnorm(1,1)
  var ~ 1/tau
  tau2 ~ dnorm(1,1)
  tau3 ~ dnorm(1,1)
  tau4 ~ dnorm(1,1)
  tau5 ~ dnorm(1,1)
  var_gamma[1] ~ 1/tau1
  var_gamma[2] ~ 1/tau2
  var_gamma[3] ~ 1/tau3
  var_gamma[4] ~ 1/tau4
  var_gamma[5] ~ 1/tau5
}

# model identification
for (i in 1:5) {
  b[i] ~ dnorm(0,1)
}
b[6] <- -1*sum(b[1:5])

```

- Data to be used
- The Rasch testlet model
- Specification of priors
- **Model identification (item side)**

## 5.10 Model Specification VII

**Model Specification**

```

testlet_rasch
model
{
  # data to be used
  for (i in 1:2) {
    for (l in 1:10) {
      r[i,l] ~ dbern(p[i,l])
    }
  }

  #IRT model
  for (i in 1:2) {
    for (l in 1:10) {
      p[i,(i-1)*10+l] ~ dexp(theta[i]+gamma[1]*b[i])
    }
  }
  for (i in 0:10) {
    p[0,(i-1)*10+l] ~ dexp(theta[i]+gamma[2]*b[i])
  }
}

for (i in 1:15) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[3]*b[i])
}
for (i in 16:20) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[4]*b[i])
}
for (i in 21:25) {
  p[1,(i-1)*10+l] ~ dexp(theta[i]+gamma[5]*b[i])
}

```

```

#specification of priors
for (i in 1:10) {
  theta[i] ~ dnorm(0,tau)
  gamma[1] ~ dnorm(0,tau1)
  gamma[2] ~ dnorm(0,tau2)
  gamma[3] ~ dnorm(0,tau3)
  gamma[4] ~ dnorm(0,tau4)
  gamma[5] ~ dnorm(0,tau5)
  tau ~ dnorm(0,1)
  tau1 ~ dnorm(1,1)
  tau2 ~ dnorm(1,1)
  tau3 ~ dnorm(1,1)
  tau4 ~ dnorm(1,1)
  tau5 ~ dnorm(1,1)
  var_gamma[1] ~ 1/tau1
  var_gamma[2] ~ 1/tau2
  var_gamma[3] ~ 1/tau3
  var_gamma[4] ~ 1/tau4
  var_gamma[5] ~ 1/tau5
  for (j in 1:5)
    b[j] ~ dnorm(0,1)
}

```

- Data to be used
- The Rasch testlet model
- Specification of priors
- **Model identification (person side)**

## **5.11 Reading in Data I**

## **5.12 The Partial Credit Testlet Model I**

## 5.13 The Partial Credit Testlet Model I

**The Partial Credit Testlet Model**

Assume four score categories: 0, 1, 2, 3  
Recode scores into 1, 2, 3, 4

The probability of getting a certain score is:

$$t_1 = \exp(\theta_j + \gamma_{jd(i)} - b_i - d_{i1}), \quad t_2 = \exp(2\theta_j + 2\gamma_{jd(i)} - 2b_i - d_{i1} - d_{i2}),$$

and

$$t_3 = \exp(3\theta_j + 3\gamma_{jd(i)} - 3b_i - d_{i1} - d_{i2} - d_{i3})$$

then:

$$pr(X_{ji} = 0) = \frac{1}{1 + t_1 + t_2 + t_3} \quad pr(X_{ji} = 1) = \frac{t_1}{1 + t_1 + t_2 + t_3}$$

$$pr(X_{ji} = 2) = \frac{t_2}{1 + t_1 + t_2 + t_3} \quad pr(X_{ji} = 3) = \frac{t_3}{1 + t_1 + t_2 + t_3}$$

## 5.14 The Partial Credit Testlet Model II

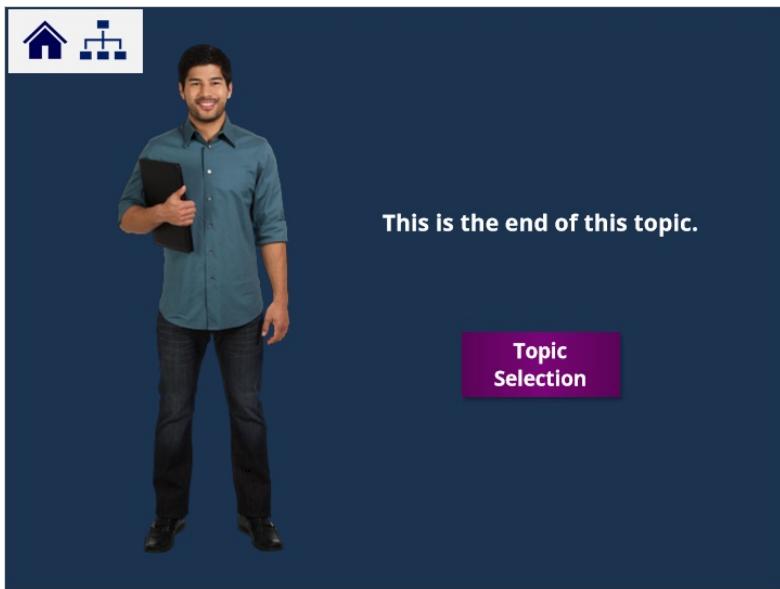
**The Partial Credit Testlet Model**

```
testlet-pcm-pisa.txt
model
{
  for (j in 1:2) {
    for (i in 1:6) {
      if (i<-response) {
        d[i] = 0;
      }
    }
  }
  # Testlet PCM model
  for (i in 1:6) {
    for (j in 1:2) {
      d[i] = 0;
      if (i>=response) {
        d[i] = 1;
      }
      if (i>=1) {
        p[i,1] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
        p[i,2] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
        p[i,3] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
        p[i,4] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
        p[i,5] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
        p[i,6] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
      }
    }
  }
  for (i in 12:19) {
    d[i] = 0;
    if (i>=response) {
      d[i] = 1;
    }
    if (i>=1) {
      p[i,1] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      p[i,2] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
      p[i,3] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      d[i] = 0;
      if (i>=1) {
        p[i,4] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
        p[i,5] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
        p[i,6] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      }
    }
  }
  for (i in 20:28) {
    d[i] = 0;
    if (i>=response) {
      d[i] = 1;
    }
    if (i>=1) {
      p[i,1] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      p[i,2] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
      p[i,3] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      d[i] = 0;
      if (i>=1) {
        p[i,4] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
        p[i,5] = exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]);
        p[i,6] = 1-exp(-exp(2*(theta[j]-2*gamma[j]-d[i]))*(1-d[i]));
      }
    }
  }
}

testlet-pcm-pisa.txt
#Specify Prior for person parameters
p[1,1] ~ dnorm(0,1)
p[1,2] ~ dnorm(0,tau1)
p[1,3] ~ dnorm(0,tau2)
p[1,4] ~ dnorm(0,tau3)
mu~dnorm(0,1)
tau1~dgamma(1,1)
tau2~dgamma(1,1)
tau3~dgamma(1,1)
#Specify Prior for item parameters
for (i in 1:6) {
  d[i,1] ~ dnorm(0,1)
  d[i,2] ~ dnorm(0,1)
  d[i,3] ~ dnorm(0,1)
  d[i,4] ~ dnorm(0,1)
  d[i,5] ~ dnorm(0,1)
  d[i,6] ~ dnorm(0,1)
}
for (i in 12:19) {
  d[i,1] ~ dnorm(0,1)
  d[i,2] ~ dnorm(0,1)
  d[i,3] ~ dnorm(0,1)
  d[i,4] ~ dnorm(0,1)
  d[i,5] ~ dnorm(0,1)
  d[i,6] ~ dnorm(0,1)
}
for (i in 20:28) {
  d[i,1] ~ dnorm(0,1)
  d[i,2] ~ dnorm(0,1)
  d[i,3] ~ dnorm(0,1)
  d[i,4] ~ dnorm(0,1)
  d[i,5] ~ dnorm(0,1)
  d[i,6] ~ dnorm(0,1)
}
```

- Testlet variances
  - Items nested with testlets as
    - Items 1 to 11---testlet 1
    - Items 12 to 19---testlet 2
    - Items 20 to 28---testlet 3
  - Testlet effect parameters follow a normal distribution with a mean of 0 an standard deviation to be estimated
  - Testlet variances follow an inverse gamma distribution.
- Model identification
  - Item side
  - Step parameters to be constrained
- Item responses
  - Starts at 1

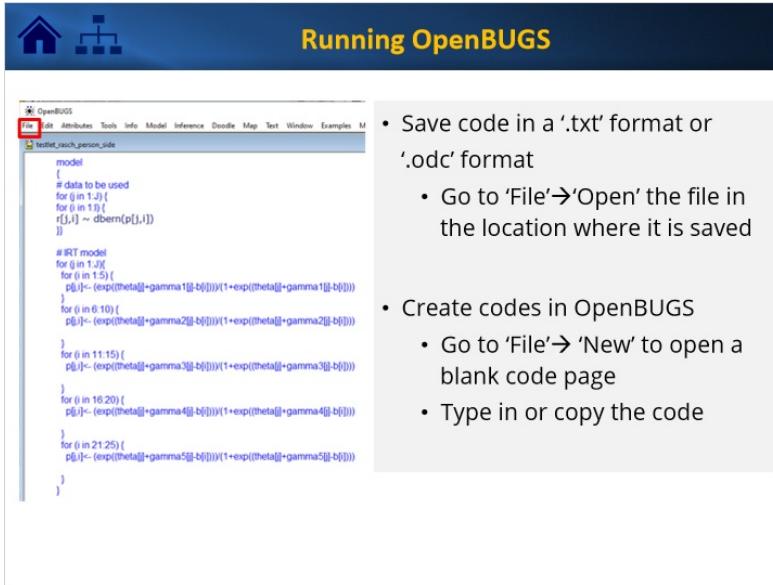
### **5.15 Bookend: Preparing the Script and Data**



### **5.16 Bookmark: Running OpenBUGS**



## 5.17 Running OpenBUGS I



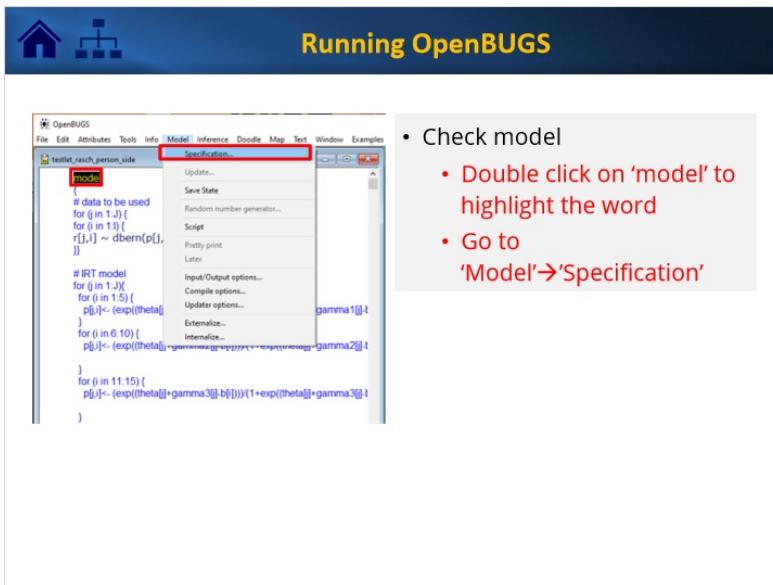
The screenshot shows the OpenBUGS software interface. The title bar says "Running OpenBUGS". The menu bar includes File, Edit, Attributes, Tools, Info, Model, Inference, Doodle, Map, Text, Window, Examples, and Help. The "File" menu is highlighted with a red box. The main window displays an R script titled "testlet\_reach\_person\_size". The script contains the following code:

```
model
{
  if data to be used
  for (j in 1:J) {
    for (i in 1:I) {
      r[j,i] ~ dbern(p[j,i])
    }
  }

  #IRT model
  for (j in 1:J)
    for (i in 1:5) {
      p[j,i] <- (exp(theta[j]+gamma[1][i]*b[i]))/(1+exp(theta[j]+gamma[1][i]*b[i]))
    }
  for (i in 6:10) {
    p[j,i] <- (exp(theta[j]+gamma[2][i]*b[i]))/(1+exp(theta[j]+gamma[2][i]*b[i]))
  }
  for (i in 11:15) {
    p[j,i] <- (exp(theta[j]+gamma[3][i]*b[i]))/(1+exp(theta[j]+gamma[3][i]*b[i]))
  }
  for (i in 16:20) {
    p[j,i] <- (exp(theta[j]+gamma[4][i]*b[i]))/(1+exp(theta[j]+gamma[4][i]*b[i]))
  }
  for (i in 21:25) {
    p[j,i] <- (exp(theta[j]+gamma[5][i]*b[i]))/(1+exp(theta[j]+gamma[5][i]*b[i]))
  }
}
```

- Save code in a '.txt' format or '.odc' format
  - Go to 'File'→'Open' the file in the location where it is saved
- Create codes in OpenBUGS
  - Go to 'File'→'New' to open a blank code page
  - Type in or copy the code

## 5.18 Running OpenBUGS II



The screenshot shows the OpenBUGS software interface. The title bar says "Running OpenBUGS". The menu bar includes File, Edit, Attributes, Tools, Info, Model, Inference, Doodle, Map, Text, Window, Examples, and Help. The "Model" menu is highlighted with a red box. A context menu is open over the word "model" in the R script. The menu options include Specification..., Update..., Save State, Random number generator..., Script, Pretty print, Input/Output options..., Compile options..., Updater options..., Externalise..., Internalise..., and Latent. The "Specification..." option is highlighted with a red box. The main window displays the same R script as in the previous screenshot.

- Check model
  - Double click on 'model' to highlight the word
  - Go to 'Model'→'Specification'

## 5.19 Running OpenBUGS III

The screenshot shows the OpenBUGS software interface with the title bar "Running OpenBUGS". The main window displays a BUGS model code for "testlet\_reach\_person\_code". A "Specification Tool" dialog box is open, containing buttons for "check model", "load data", "compile", "num of chains", "load mcmc", and "for chain". The "check model" button is highlighted with a red box. Below the dialog box, a message says "Model is correctly stated".

- Check model
  - Double click on 'model' to highlight the word
  - Go to 'Model'→'Specification'
  - Click on 'check model'

## 5.20 Running OpenBUGS V

The screenshot shows the OpenBUGS software interface with the title bar "Running OpenBUGS". The main window displays a BUGS model code for "testlet\_reach\_person\_code". A "Specification Tool" dialog box is open, containing buttons for "check model", "load data", "compile", "num of chains", "load mcmc", and "for chain". The "check model" button is highlighted with a red box. Below the dialog box, a message says "Model is correctly stated". At the bottom left of the main window, there is a "Data loader" button highlighted with a red box.

- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'

## 5.21 Running OpenBUGS VI

Running OpenBUGS

- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'
- Type in 'num of chains' '2'
- Click on 'compile'

## 5.22 Running OpenBUGS VII

Running OpenBUGS

- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'
- Type in 'num of chains' '2'
- Click on 'compile'
- Click on 'gen inits' to generate initial values

## 5.23 Running OpenBUGS VIII

**Running OpenBUGS**

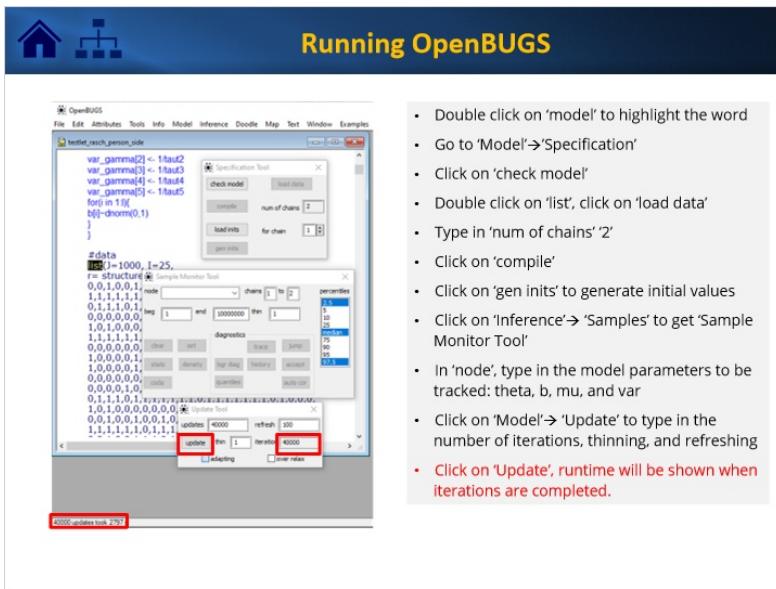
- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'
- Type in 'num of chains' '2'
- Click on 'compile'
- Click on 'gen inits' to generate initial values
- Click on 'Inference'→'Samples' to get 'Sample Monitor Tool'
- In 'node', type in the model parameters to be tracked: theta, b, mu, and var

## 5.24 Running OpenBUGS IX

**Running OpenBUGS**

- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'
- Type in 'num of chains' '2'
- Click on 'compile'
- Click on 'gen inits' to generate initial values
- Click on 'Inference'→'Samples' to get 'Sample Monitor Tool'
- In 'node', type in the model parameters to be tracked: theta, b, mu, and var
- Click on 'Model'→'Update' to type in the number of iterations, thinning, and refreshing

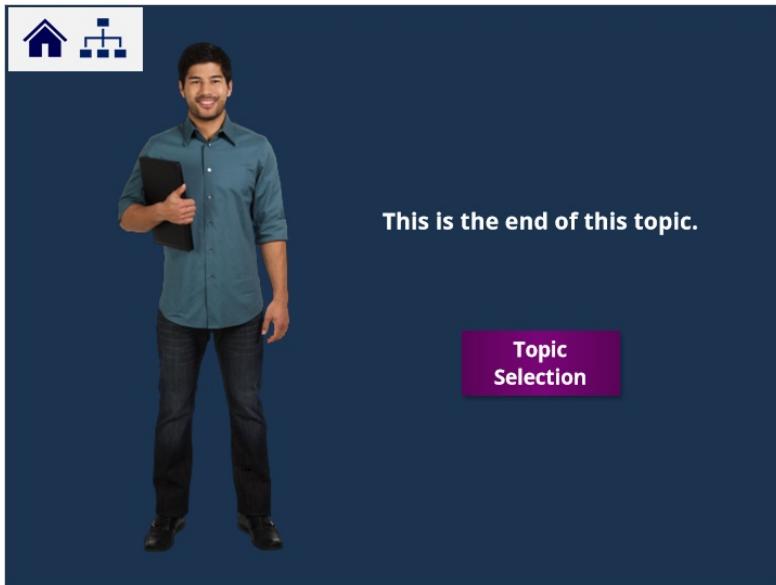
## 5.25 Running OpenBUGS X



The screenshot shows the OpenBUGS X software interface. At the top, there's a navigation bar with icons for Home, Model, Inference, Doodle, Map, Text, Window, and Examples. Below the navigation bar, the main window has two tabs: 'Specification Tool' and 'Sample Monitor Tool'. The 'Specification Tool' tab is active, displaying a text editor with R code for a model named 'testlet\_reach\_person\_size'. The 'Sample Monitor Tool' tab shows a hierarchical tree of nodes being tracked, with parameters like theta, b, mu, and var highlighted in red boxes. A status bar at the bottom indicates '40000 update took 200s'.

- Double click on 'model' to highlight the word
- Go to 'Model'→'Specification'
- Click on 'check model'
- Double click on 'list', click on 'load data'
- Type in 'num of chains' 2
- Click on 'compile'
- Click on 'gen inits' to generate initial values
- Click on 'Inference'→ 'Samples' to get 'Sample Monitor Tool'
- In 'node', type in the model parameters to be tracked: theta, b, mu, and var
- Click on 'Model'→ 'Update' to type in the number of iterations, thinning, and refreshing
- Click on 'Update', runtime will be shown when iterations are completed.

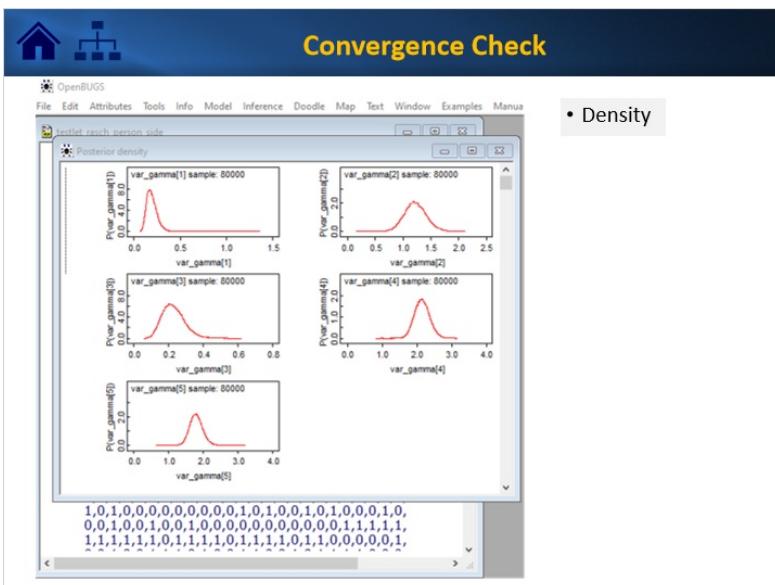
## 5.26 Bookend: Running OpenBUGS



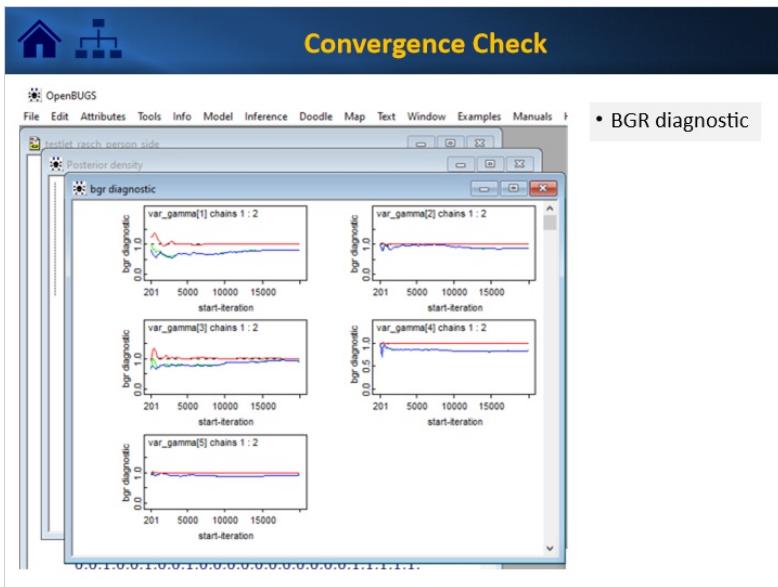
## **5.27 Bookmark: Interpreting Output**



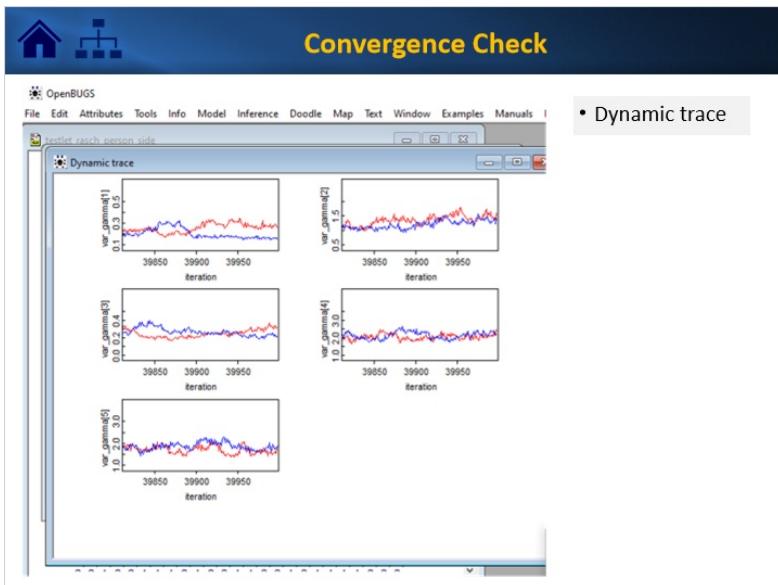
## 5.28 Convergence Check I



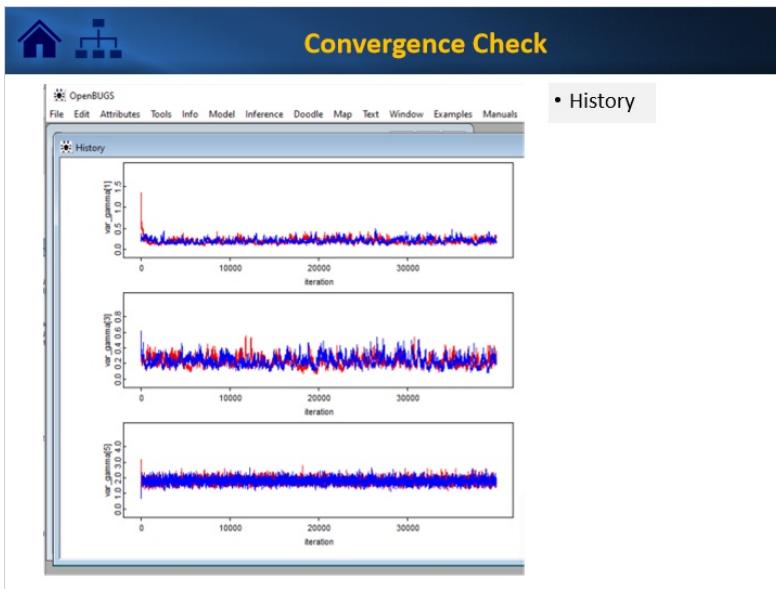
## 5.29 Convergence Check II



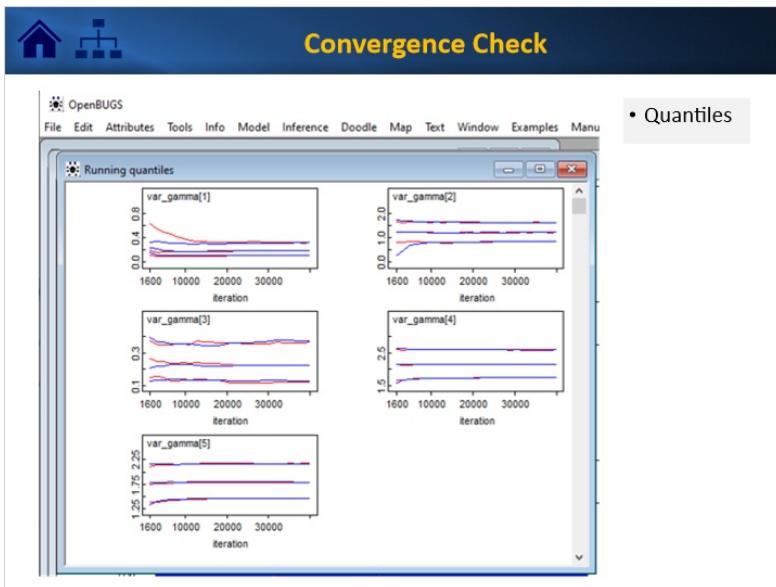
## 5.30 Convergence Check III



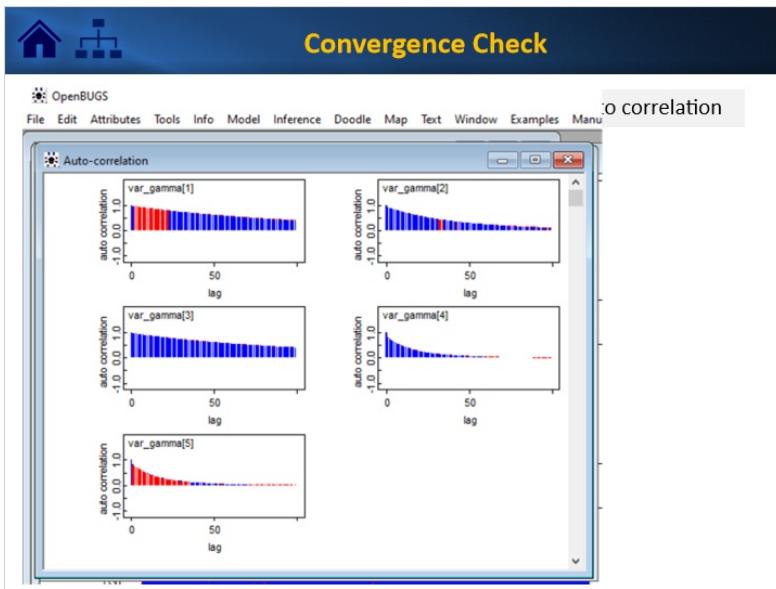
### 5.31 Convergence Check IV



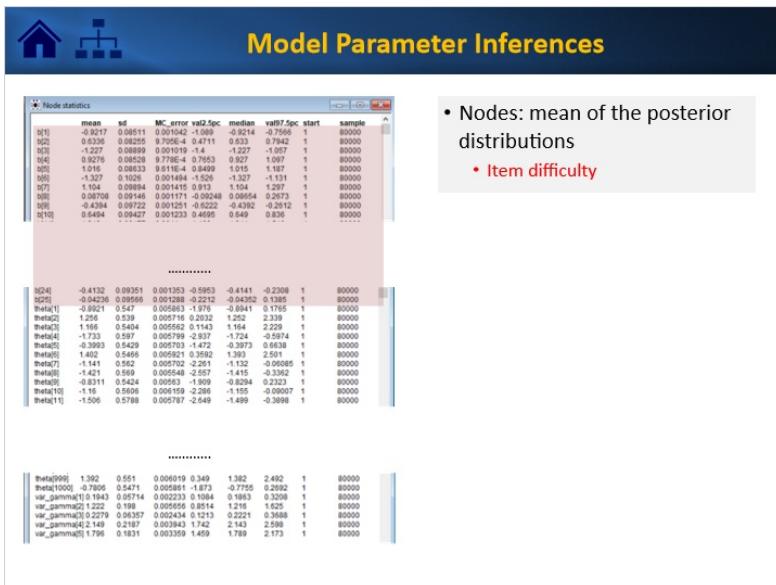
### 5.32 Convergence Check V



### 5.33 Convergence Check VI



### 5.34 Model Parameter Inferences I



## 5.35 Model Parameter Inferences II

**Model Parameter Inferences**

• Nodes: mean of the posterior distributions

- Item difficulty
- Person ability

	mean	sd	MC_error	val2.spc	median	val97.spc	start	sample
b[1]	-0.9217	0.08511	0.001042	-1.089	-0.9214	-0.7566	1	80000
b[2]	0.9217	0.08511	0.001042	0.7058	0.9211	0.7563	1	80000
b[3]	-1.227	0.08899	0.001916	-1.227	-1.227	-1.057	1	80000
b[4]	0.9276	0.08528	0.001984	0.927	0.927	1.097	1	80000
b[5]	1.104	0.08894	0.001415	1.104	1.104	1.197	1	80000
b[6]	-1.327	0.1028	0.001484	-1.327	-1.327	-1.131	1	80000
b[7]	1.104	0.08894	0.001415	0.913	1.104	1.297	1	80000
b[8]	0.9276	0.08528	0.001984	0.927	0.927	1.097	1	80000
b[9]	-0.4394	0.09722	0.001251	-0.5222	-0.4392	-0.2612	1	80000
b[10]	0.6494	0.09427	0.00233	0.6995	0.649	0.836	1	80000
.....								
b[24]	-0.4132	0.09351	0.001053	-0.9593	-0.4141	-0.2308	1	80000
b[25]	-0.4242	0.09566	0.001288	-0.2121	-0.4352	0.1385	1	80000
theta[1]	1.256	0.539	0.005716	0.2932	1.252	2.339	1	80000
theta[2]	1.168	0.5404	0.005562	0.1143	1.164	2.229	1	80000
theta[3]	-1.227	0.539	0.005716	-0.2932	-1.227	0.057	1	80000
theta[4]	0.9276	0.08528	0.001984	0.9253	0.927	1.097	1	80000
theta[5]	-0.3993	0.5429	0.005703	-1.472	-0.3973	0.6438	1	80000
theta[6]	1.402	0.5468	0.005521	0.3992	1.393	2.501	1	80000
theta[7]	-1.421	0.559	0.005548	-2.557	-1.421	0.057	1	80000
theta[8]	-1.421	0.569	0.005548	-2.557	-1.415	-0.3352	1	80000
theta[9]	-0.8311	0.5424	0.005563	-1.809	-0.8294	0.2323	1	80000
theta[10]	-1.506	0.5788	0.005703	-1.509	-1.505	-0.9807	1	80000
theta[11]	-1.506	0.5788	0.005787	-2.649	-1.499	-0.3898	1	80000
.....								
theta[999]	1.392	0.551	0.0001919	0.349	1.382	2.482	1	80000
theta[1000]	-0.7805	0.5471	0.005881	-1.873	-0.7755	0.2692	1	80000
var_gamma[1]	0.1943	0.05714	0.002233	0.1084	0.1883	0.3298	1	80000
var_gamma[2]	0.2279	0.05357	0.002434	0.1013	0.2221	0.3688	1	80000
var_gamma[3]	0.2279	0.06357	0.002434	0.1213	0.2221	0.3688	1	80000
var_gamma[4]	0.2149	0.2187	0.003943	1.742	2.143	2.598	1	80000
var_gamma[5]	0.1831	0.003359	1.459	1.789	2.173	1	80000	

## 5.36 Model Parameter Inferences III

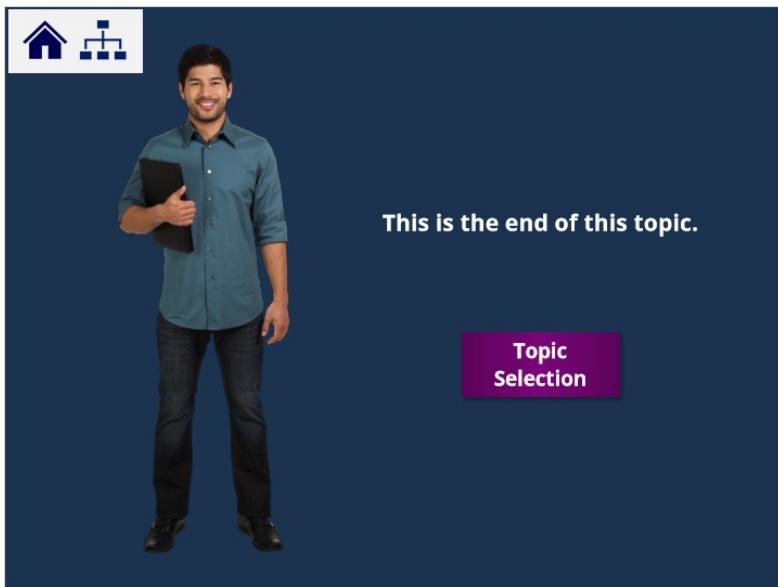
**Model Parameter Inferences**

• Nodes: mean of the posterior distributions

- Item difficulty
- Person ability
- Variance of the testlet effects

	mean	sd	MC_error	val2.spc	median	val97.spc	start	sample
b[1]	-0.9217	0.08511	0.001042	-1.089	-0.9214	-0.7566	1	80000
b[2]	0.9217	0.08511	0.001042	0.7058	0.9211	0.7563	1	80000
b[3]	-1.227	0.08899	0.001916	-1.227	-1.227	0.057	1	80000
b[4]	0.9276	0.08528	0.001984	0.9253	0.927	1.097	1	80000
b[5]	1.104	0.08894	0.001415	0.913	1.104	1.197	1	80000
b[6]	-1.327	0.1028	0.001484	-1.526	-1.327	-1.131	1	80000
b[7]	1.104	0.08894	0.001415	0.913	1.104	1.297	1	80000
b[8]	0.9276	0.08528	0.001984	0.9253	0.927	1.097	1	80000
b[9]	-0.4394	0.09722	0.001251	-0.5222	-0.4392	-0.2612	1	80000
b[10]	0.6494	0.09427	0.00233	0.6995	0.649	0.836	1	80000
.....								
b[24]	-0.4132	0.09351	0.001053	-0.9593	-0.4141	-0.2308	1	80000
b[25]	-0.4242	0.09566	0.001288	-0.2121	-0.4352	0.1385	1	80000
theta[1]	1.256	0.539	0.005716	0.2932	1.252	2.339	1	80000
theta[2]	1.168	0.5404	0.005562	0.1143	1.164	2.229	1	80000
theta[3]	-1.227	0.539	0.005716	-0.2932	-1.227	0.057	1	80000
theta[4]	0.9276	0.08528	0.001984	0.9253	0.927	1.097	1	80000
theta[5]	-0.3993	0.5429	0.005703	-1.472	-0.3973	0.6438	1	80000
theta[6]	1.402	0.5468	0.005521	0.3992	1.393	2.501	1	80000
theta[7]	-1.421	0.559	0.005548	-2.557	-1.421	-0.3352	1	80000
theta[8]	-1.421	0.569	0.005548	-2.557	-1.415	-0.3352	1	80000
theta[9]	-0.8311	0.5424	0.005563	-1.809	-0.8294	0.2323	1	80000
theta[10]	-1.506	0.5788	0.005703	-2.649	-1.499	-0.3898	1	80000
theta[11]	-1.506	0.5788	0.005787	-2.649	-1.499	-0.3898	1	80000
.....								
theta[999]	1.392	0.551	0.0001919	0.349	1.382	2.482	1	80000
theta[1000]	-0.7805	0.5471	0.005881	-1.873	-0.7755	0.2692	1	80000
var_gamma[1]	0.1943	0.05714	0.002233	0.1084	0.1883	0.3298	1	80000
var_gamma[2]	0.2279	0.05357	0.002434	0.1013	0.2221	0.3688	1	80000
var_gamma[3]	0.2279	0.06357	0.002434	0.1213	0.2221	0.3688	1	80000
var_gamma[4]	0.2149	0.2187	0.003943	1.742	2.143	2.598	1	80000
var_gamma[5]	0.1831	0.003359	1.459	1.789	2.173	1	80000	

### 5.37 Bookend: Interpreting Output

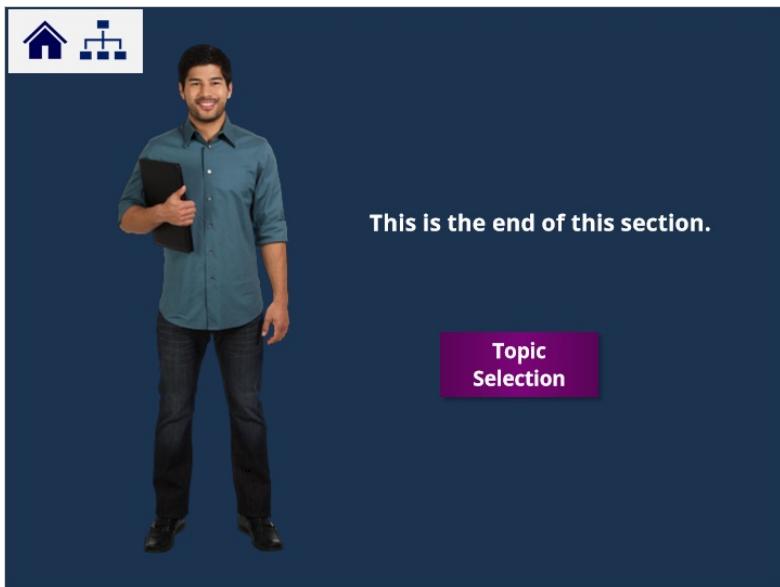


### 5.38 Summary

A slide with a dark blue header bar. On the left of the bar are icons for a house and a stack of books. In the center, the word "Summary" is written in yellow. The main body of the slide contains a bulleted list of topics covered in the chapter:

- Preparing data and model
  - ✓ Defining testlet response model
  - ✓ Setting priors
- Point-and-click interface to OpenBUGS
- Reading output
  - ✓ Diagnostics such as convergence checks
  - ✓ Interpreting estimated model parameter values

### **5.39 Bookend: Section 5**



### **5.40 Module Cover (END)**

