Study of Structural Equation Models and Their Application to Fitchburg Middle School Data

by

Jonathan Charles Legare

A Project Report

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Applied Statistics

January 13, 2009

Approved by

____

Professor Ryung Kim, Capstone Advisor

____

Professor Bogden Vernescu, Department Head

Abstract

Structural equation models combine factor analysis models and multivariate regression models to estimate associations between observed variables and unobserved variables. The main achievement of this Capstone Project is the understanding of structural equation models   and application of the models to real-world data. In this report, we reviewed structural equation models and several prerequisite topics. We performed a simulation study to compare maximum likelihood structural equation model estimation versus two-stage sequential estimation using multiple linear regression and maximum likelihood factor analysis. The simulation study confirmed that confidence intervals produced by structural equation models are valid and those obtained by two-stage sequential estimation are largely inaccurate. We applied structural equation models to an educational data comparing the efficacy of teaching conditions on learning scientific inquiry skills among 177 middle school students in Fitchburg, Massachusetts using a computer simulated science microworld. Application of structural equation models to the educational data showed that there were no significant differences in test score gains between three learning conditions, while controlling for latent factors measured by survey responses.

Acknowledgements

A healthy portion of my last eight years have been spent in Stratton Hall at WPI as a full-time undergraduate in mathematics and as a part-time graduate student in applied statistics. Many fellow students and professors have provided motivation, guidance, and even friendship. Thanks to all of you.

In particular, I would like extend my sincere thanks to Professor Joseph D. Petruccelli. You were my academic advisor throughout my entire academic career at WPI, advisor to my undergraduate major qualifying project, and instructor of several graduate and undergraduate statistics courses. Thank you for your patience, for your motivation, for your mentorship, and for combining your sense of academic professionalism with your sense of humor. And thanks for the pizza too!

To my capstone project advisor, Professor Ryung S. Kim: Your patience, academic curiosity, and high expectations made this project successful; it was a rewarding experience working with you. Thank you for your time each week counseling me. Thank you for your guidance, especially around simulation studies and estimating interaction terms in structural equations, which much of this project depended on.

To my capstone co-advisor, Professor Janice Gobert: Thank you for co-advising for my capstone project, which was neither expected nor required. Your research questions motivated the use of factor analysis and structural equations, and you also provided the data analyzed in this report. Thank you for meeting on Friday afternoons, and thank you for your guidance in this analysis.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

SECTION 1: INTRODUCTION

Structural equation models (SEM) combine factor analysis and multivariate regression models to estimate associations between observable variables and unobservable (latent) variables. The main achievement of this Capstone Project is the understanding of structural equation models and application of the models to real-world data.

Section 2 introduces the Assistments project, the source of our real-world data. The Assistments team measured the scientific inquiry skills of middle school students, specifically their understanding of the control for variables strategy. The data the team collected are used throughout this report. Section 3 reviews topics prerequisite to the understanding and application of structural equation models. Topics covered are path diagrams, factor models, parallel analysis and orthogonal rotation. These tool were applied to the Assistments data. Section 4 provides a review and real-world explanation of structural equation models. In this section we evaluated the estimation accuracy of SEM through simulation and then we analyzed the Assistments data using a bivariate structural model.

Through a simulation study we confirmed that the validity of confidence intervals produced by SEM are valid and those obtained by two-stage sequential estimation are largely inaccurate.. Finally our application of structural equation models to the Assistments data showed no significant differences in test score gains between learning conditions, while controlling for latent factors.

SECTION 2: THE ASSISTMENTS EXPERIMENT

In this section, we introduce the Assistments project, which focuses on the scientific inquiry skills of middle school students. We define scientific inquiry skills and the control for variables strategy, and then explain an experiment conducted by the Assistments team which measured these skills. Finally we report the results from our basic analysis of the experimental data, which tested the randomization of treatment groups, and also tested for learning gains and treatment effects.

The Assistments Project

"Computer Science Assistant Professor Neil Heffernan of Worcester Polytechnic Institute (WPI) and his colleagues at Carnegie Mellon University (CMU) are the developers of the Assistments program (www.assistments.org). This intelligent tutoring system is a web-based system for Math that provides example problems and reacts to students' inputs in real time. Math Assistments is presently used by students and teachers in Massachusetts with the goal of improving students' math learning. One of the measures used to assess student improvement is the Massachusetts Comprehensive Assessment System (MCAS) score for Math. There are various types of mathematical concepts covered in the Assistments problems to improve MCAS scores." (Richardson, 2008)

Social Sciences Professor Janice Gobert of Worcester Polytechnic Institute (WPI) is the principal investigator for this project. The following description is paraphrased from her annual report. "The long-term goal of this project is to improve inquiry skills in middle school, so that students can transfer this knowledge to other science topics, and ideally to real-world problems and improve upon content learning as well. Research suggests that these skills should be developed during middle school. The extent to which inquiry skills can be learned and then transferred to learn new domains will be determined as part of the project." For more information of the Assistments system, refer to (Gobert) and (Richardson, 2008).

Scientific Inquiry and the Control for Variables Strategy

The method of **scientific inquiry** is the investigative process which scientists use to understand the natural world. Scientific inquiry skills are important for conducting experiments and making decisions in real-world situations. They are not developed naturally, and many student lack these vital skills. The National Science Education Standards (NSES, 1996) is a framework for science education in primary and secondary schools in the United States. They defines scientific inquiry skills as follows:

"Scientific inquiry refers to the diverse ways in which scientists study the natural world and propose explanations based on the evidence derived from their work. Inquiry also refers to the activities of students in which they develop knowledge and understanding of scientific ideas, as well as an understanding of how scientists study the natural world."

"Inquiry is a multifaceted activity that involves making observations; posing questions; examining books and other sources of information to see what is already known; planning investigations; reviewing what is already known in light of experimental evidence; using tools to gather, analyze, and interpret data; proposing answers, explanations, and predictions; and communicating the results. Inquiry requires identification of assumptions, use of critical and logical thinking, and consideration of alternative explanations. Students will engage in selected aspects of inquiry as they learn the scientific way of knowing the natural world, but they also should develop the capacity to conduct complete inquiries." (NSES, 1996 p.23)

The **control for variables strategy** is a strategy under the broader inquiry strand of conducting scientific investigations. It involves measuring the effect on a response variable by manipulating one independent variable and holding all other independent variables constant. The strategy investigates the relationship between two variables at a time.

Consider an experiment where a ball is rolled down an inclined plane. The response variable is the distance the ball rolls. Many variables could affect the distance the ball rolls, such as the surface roughness of the inclined plane, the angle of the inclined plane, or the type of ball. To measure the effect of ramp steepness using the control for variables strategy, the experimenter would vary the ramp steepness while holding all other variables constant.

Experimental Design and Procedure

WPI social science Professor Janice Gobert and WPI computer science graduate student Michael Sao Pedro designed and conducted an experiment which explored the scientific inquiry skills of middle-school students. Specifically it focused on their understanding of the control for variables strategy, in which one measures the effect of one variable while holding all other variables constant.

Participants

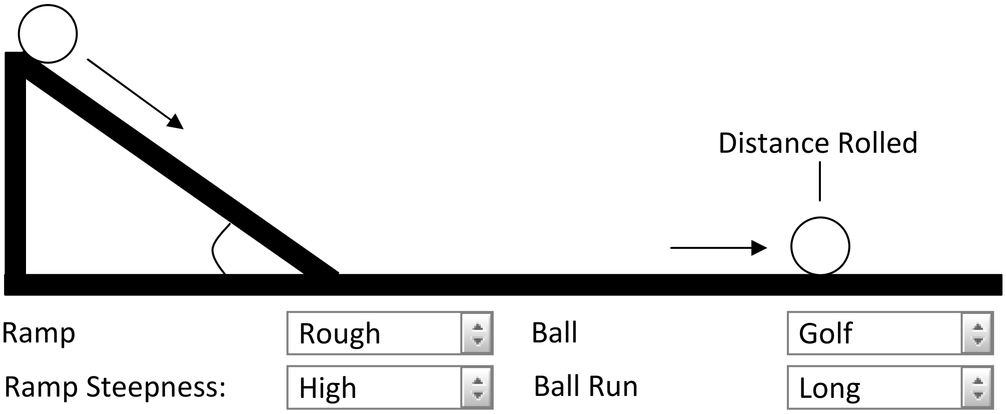
The study participants were students chosen from a Central Massachusetts middle school. This school was decided to be a good candidate for the study because it is ethnically diverse, has large numbers of English language learner (ELL) students, has a large proportion of students on free or assisted lunch programs, and had a substantial number of below-proficient scores on the science portion of the Massachusetts Comprehensive Assessment System (MCAS) exam.

A total of 177 students were selected from the 7th and 8th grades, with 98 students in the 7th grade and 79 students in the 8th grade. Students from each grade were chosen from 5 different class periods but with the same teacher for students in each grade level. Selecting from multiple periods was done such that students were chosen from a range of class levels, from low performing with paraprofessional help to average and above average performing.

Materials

A ramp apparatus (see Figure 1) was simulated as a **microworld** by the Assistments team. A microworld is a tiny computer simulated world in which a student can explore alternatives, test hypotheses, and discover scientific principles. The microworld allowed students to change the steepness, run length, type of ball and ramp surface. The simulation includes two ramps to allow for direct side-by-side comparison. The student can choose the four parameters for each ramp.

Figure 1: Ramp Apparatus



Ramp: Rough
Ball: Golf
Ramp Steepness: High
Ball Run: Long

Pre-tests Surveys and Post-tests

Students were given two pre-tests to measure their scientific inquiry skills and the control for variables strategy, specifically. Two surveys were administered to measure their understanding of scientific models (SUMS) and their perseverance and passion for long-term goals (GRIT). Both surveys are described in more detail in the next section. The students were also given two post-tests to measure the gain in understanding.

The **inquiry pre-test** and **inquiry post-test** measure inquiry skills, which include formulating hypotheses, data interpretation, conducting experiments (including the control for variables strategy) and replicability of experiments. The **ramp experiment pre-test** requests students to design an experiment to determine how steepness and run length affect the distance rolled using the two ramps in the microworld. The **ramp experiment post test** focuses on all four of the available variables in the ramp microworld, unlike the pre-test which is concerned only with steepness and run length.

The term **SUMS** is an acronym for Students' Understanding of Models in Science. The survey consists of twenty-seven questions on a five-point Likert scale. It was developed by David F. Treagust et al (2002) with the specific objective of learning more about students' understanding of what a model is, and the role of models in science. Briefly, it measures students' epistemologies of models on a continuum from simple, i.e., models are exact replicas, to intermediate, i.e., models are explanatory tools, to complex, i.e., models are tools for hypothesis-generation and scientific prediction.

The **grit survey** consists of thirteen questions on a five-point Likert scale. The concept of grit was developed by Angela Duckworth (2007), to help address the question: Why do some individuals accomplish more than others of equal intelligence? She defines grit as: "perseverance and passion for long-term goals. Grit entails working strenuously toward challenges, maintaining effort and interest over years despite failure, adversity, and plateaus in progress. The gritty individual approaches achievement as a marathon: his or her advantage is stamina. Whereas disappointment or boredom signals to others that it is time to change trajectory and cut losses, the gritty individual stays the course." (Duckworth, 2007 pp. 1087-1088)

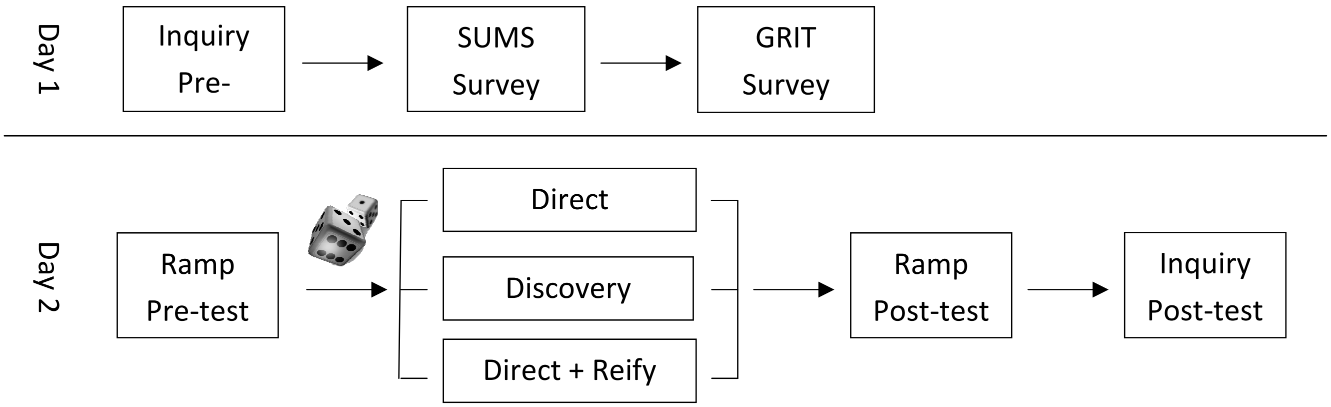
Experimental Procedure

The following (simplified) procedure was implemented by the Assistments team. Students were first given the two pre-tests to measure their scientific inquiry skills and initial understanding of the ramp experiment. Two surveys were administered to measure their understanding of scientific models (SUMS) and their perseverance and passion for long-term goals (GRIT).

Each student was then randomly assigned by the Assistments system to one of three treatment groups. The “discovery” learning condition in which students explored a ramp microworld to “discover” the control for variables strategy. The “direct” learning condition in which students were given the microworld (but did not explore it directly) and were given explicit instruction about the control for variables strategy in the form of a textual description of it. The “direct plus reify” learning condition, in which students were given a description of the control for variables strategy (as in b, above) but were also asked, at specific points, to provide written explanations of their understanding of the control for variables strategy.

After learning about the control for variables strategy, each student was given the two post-tests to measure their scientific inquiry skills and final understanding of the ramp experiment. For each student we computed the "gain score" as post-test score minus pre-test score. We are interested in the effect of the three learning conditions (treatments) and the students' survey responses on the gain scores.

Figure 2: Experimental Procedure



Note: the above diagram is based on a presentation by Michael Sao Pedro, used with permission.

SECTION 3: PATH DIAGRAMS AND FACTOR ANALYSIS

In this section we review topics prerequisite to understanding and applying structural equation models. These topics include path diagrams, factor models, parallel analysis and orthogonal rotation. Path diagrams allow for the visualization of systems of equations, such as regression models and factor models. Factor analysis is a method for measuring unobservable factors by extracting information common to several observed variables. Parallel analysis can be used to determine the number of factors to model and orthogonal factor rotation may aid in the interpretation of a factor solution. Then, we apply these tools to the Assistments data.

Path Diagrams

A path diagram is a pictorial representation of a system of simultaneous equations. Path diagrams illustrate relationships and dependencies between variables and distinguish observable, measured variables from unobservable, theoretical latent variables. These diagrams are a fundamental tool of structural equation modeling.

A path diagram contains many symbols: A rectangle or a square represents an observed variable and a circle or an ellipse represents an unobserved or latent variable. An unenclosed variable represents an unobservable measurement error or disturbance. Straight arrows signify causality or dependence, with letters above the lines denoting regression coefficients. Two-headed arrows signify an association between two variables.

Figure 3: Path Diagrams

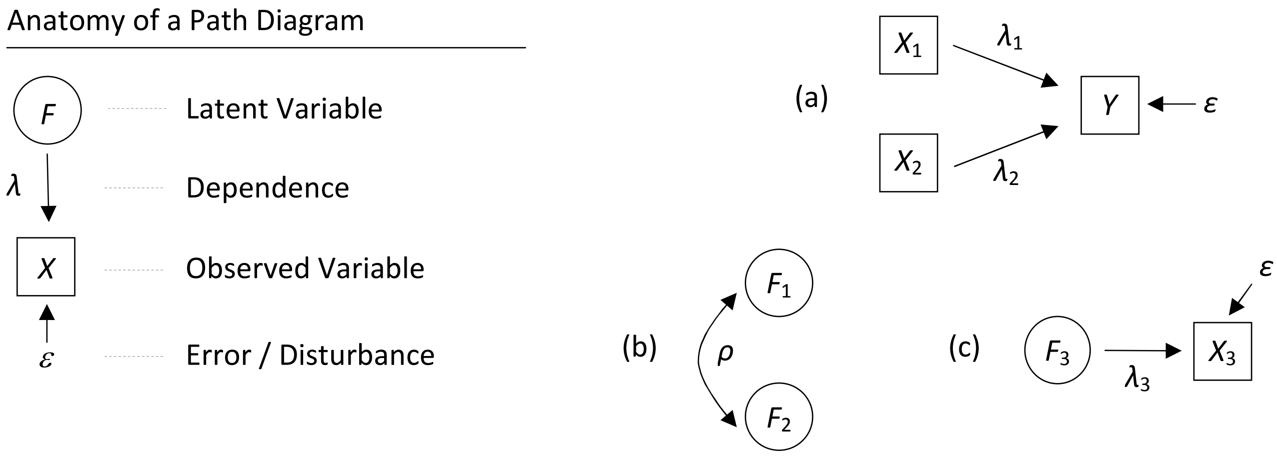


Figure 3 contains a labeled path diagram, emphasizing the main path diagram components used throughout this Project Report. The path diagram in (a) is that of a simple linear regression model with equation $Y = \lambda_1 X_1 + \lambda_2 X_2 + \varepsilon$ with observed independent and dependent variables. The diagram in (b) depicts two latent variables sharing an association; and the diagram in (c) shows an observed variable dependent on a latent variable. For a more thorough treatment of path diagrams, see (Bollen, 1989).

Factor Analysis and The Factor Model

Factor analysis is a statistical method for studying the variability among many observed intercorrelated variables in terms of fewer unobserved latent variables called factors. Factor analysis condenses many highly correlated observed variables into fewer components by analyzing patterns in the correlation matrix of observed variables. The resulting factors may represent theoretical variables of interest that are otherwise unobservable, and allow for hypothesis testing on these theoretical variables (Pohlmann, p.14).

The factor model is written as:

$$\underset{(p\times 1)}{\mathbf{X}_i} = \underset{(p\times 1)}{\boldsymbol{\mu}} + \underset{(p\times m)}{\mathbf{L}} \underset{(m\times 1)}{\mathbf{F}_i} + \underset{(p\times 1)}{\boldsymbol{\varepsilon}_i}$$

where **X** is a random vector with p variables, **F** is a vector of m elements called factors with $m < p$, **L** is a $p \times m$ matrix of unknown coefficients called factor loadings, and **ε** is a vector of p elements called errors. In the literature, **F** is usually called the common factor, and **ε** is usually called the specific factor; these two vectors are assumed to be uncorrelated, and the errors **ε** are assumed to be uncorrelated with each other.

Let $\ell_{i,j}$ denote the elements of **L** so that $x_i = \mu_i + \sum_{j=1}^{m} \ell_{i,j} F_j + \varepsilon_i$, then the expanded factor model is

$$\begin{aligned}
x_1 &= \mu_1 + \ell_{1,1} F_1 + \ell_{1,2} F_2 + \cdots + \ell_{1,j} F_j + \varepsilon_1 + \cdots + \ell_{1,m} F_m + \varepsilon_1 \\
x_2 &= \mu_2 + \ell_{2,1} F_1 + \ell_{2,2} F_2 + \cdots + \ell_{2,j} F_j + \varepsilon_2 + \cdots + \ell_{2,m} F_m + \varepsilon_2 \\
&\ \vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\
x_p &= \mu_p + \ell_{p,1} F_1 + \ell_{p,2} F_2 + \cdots + \ell_{p,j} F_j + \varepsilon_p + \cdots + \ell_{p,m} F_m + \varepsilon_p
\end{aligned}$$

Communality, h^2, is the proportion of variance in **X** explained by each of the m factors in **F**. The remaining proportion of variance, ψ_i, is due to error. The proportion of the total variance due to each of the m factors, v_j, can be estimated from the factor loadings and the sample variances.

$$h_i^2 = \sum_{j=1}^{m} \ell_{i,j}^2 \qquad \psi_i = \operatorname{var}(\mathbf{x}_i) - \sum_{j=1}^{m} \ell_{i,j}^2 \qquad v_j = \sum_{i=1}^{p} \hat{\ell}_{i,j}^2 \Big/ \sum_{i=1}^{p} \operatorname{var}(\mathbf{x}_i)$$

To estimate the factor loadings in **L**, we use the method of maximum likelihood, which assumes multivariate normality. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be n observations on p variables sampled from a multivariate normal distribution, with mean **μ** and covariance matrix **Σ**.

$$\underset{(p\times p)}{\boldsymbol{\Sigma}} = \underset{(p\times m)}{\mathbf{L}} \underset{(m\times p)}{\mathbf{L}^{\mathrm{T}}} + \underset{(p\times p)}{\boldsymbol{\psi}}, \quad \mathrm{E}(\mathbf{F}) = \mathbf{0}, \quad \mathrm{Cov}(\mathbf{F}) = \underset{(m\times m)}{\mathbf{I}}, \quad \mathrm{E}(\boldsymbol{\varepsilon}) = \mathbf{0}, \quad \mathrm{Cov}(\boldsymbol{\varepsilon}) = \underset{(p\times p)}{\boldsymbol{\psi}} = \begin{bmatrix} \psi_1 & 0 & \cdots & 0 \\ 0 & \psi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \psi_p \end{bmatrix}$$

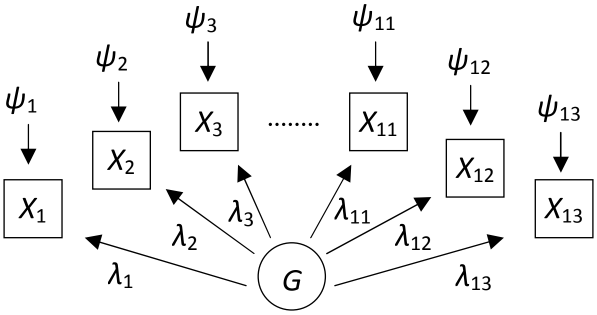
The older, traditional methods of factor analysis relied on eigendecomposition of the sample correlation matrix or a modified sample correlation matrix. The method of maximum likelihood estimation produces better estimates than these traditional methods of factor analysis. It also allows for hypothesis tests concerning the number of common factors, as well as standard error and confidence interval estimates for factor loadings and factor correlations. For more a thorough explanation of the maximum likelihood method of factor analysis, refer to (Johnson, 1999).

The factor model may be better understood using an example. Consider the grit survey data mentioned earlier, where students responded to thirteen survey questions. The factor model for this survey response data is written as

$$\mathbf{X}_i = \mathbf{L} \times G_i + \boldsymbol{\psi}_i$$

where $\mathbf{X}_i$ denotes the i^th^ student's survey responses, G denotes the grit factor which is related to **X** through the factor loadings matrix **L.** The vector of specific errors is denoted by $\boldsymbol{\psi}$. The factor model essentially transforms the thirteen survey responses into one "grit score".

Figure 4: Path Diagram of a Grit Factor Model



The grit factor model is illustrated in the path diagram in Figure 4. The path diagram contains thirteen observed survey responses, X_1, …, X_{13}. The survey responses are related to the grit factor through the factor loadings λ_1, …, λ_{13}. The specific errors ψ_1, …, ψ_{13} represent the variation in the survey responses not explained by the grit factor. Note that G is one-dimensional, so the factor loadings matrix **L** is a vector.

Orthogonal Factor Rotation

Factor rotation is a powerful tool in the factor analysis toolbox that aides in the interpretation of underlying factor structures. For reasons that will become apparent later, we did not perform factor rotation in our analysis of the Assistments data; however the procedure is nonetheless worth covering in this Report.

The factor coefficients λ_1, …, λ_p obtained through maximum likelihood estimation relate the m unobservable factors to the p observed variables. The magnitude of each coefficient offers interpretation about the relationship between the observed variables and the factors. If the factor coefficients have a complicated or un-interpretable structure, then we may **rotate** our factors through an orthogonal transformation of the factor coefficients.

Let **L** be the $p \times m$ matrix of estimated factor coefficients and let **T** be an $m \times m$ orthogonal transformation matrix with the property that $\mathbf{T}\,\mathbf{T}^\mathrm{T} = \mathbf{T}^\mathrm{T}\,\mathbf{T} = \mathbf{I}$. Then $\mathbf{L}^* = \mathbf{L}\,\mathbf{T}$ is a p x m matrix of rotated factor loadings. The orthogonal rotation preserves the maximum likelihood estimated correlation matrix, as well as the communalities and specific errors.

There are several methods for choosing the orthogonal rotation matrix **T**. The most commonly employed method of orthogonal rotation is called **varimax rotation**. The method involves an orthogonal rotation of the factor axes that maximizes the variances of the squared factor loadings. Varimax rotation produces a factor solution where each factor has few large coefficients and a many of small coefficients. Additionally, each observed variable tends to be associated with one factor.

In mathematical notation, varimax rotation searches for the orthogonal matrix **T** which maximizes the following quantity:

$$\sum_{j=1}^{m}\left\{\frac{1}{p}\sum_{i=1}^{p}\left[\frac{\ell_{i,j}^{2}}{h_i}\right]^2-\left[\frac{1}{p}\sum_{i=1}^{p}\frac{\ell_{i,j}^{2}}{h_i}\right]^2\right\}$$

Factor rotation is possible with two or more factors. A single factor structure cannot be rotated.

Determining The Number of Factors with Parallel Analysis

The purpose of exploratory factor analysis is to examine a collection of *p* observed variables to ascertain the underlying factor structure. In exploratory factor analysis, one first determines the number of underlying factors *m*, and then one estimates and interprets those factors. This section describes a modern method for determining the number of factors, *m*, which must be known prior to estimating the factor loadings.

The traditional methods for determining the number of factors are fairly subjective and should be avoided. Modern methods available today are considerably less subjective; one such method is an objective resampling technique called **parallel analysis**. There are multiple methods of parallel analysis; we use the **false discovery rate** (FDR) method.

In multiple hypothesis testing, the false discovery rate is the expected proportion of rejected null hypotheses which are erroneously rejected (Benjamini, 1995). The false discovery method is less conservative and more powerful than older family-wise error rate methods of multiple hypothesis testing, such as Bonferroni and bootstrapping. The false discovery rate method of parallel analysis is based on this concept from multiple hypothesis testing.

In traditional methods of factor analysis, eigenvectors of the sample correlation matrix correspond to factors, and the eigenvalues correspond to the variance explained by those factors. In parallel analysis the eigenvalues of the observed correlation matrix are compared against eigenvalues of sample correlation matrices obtained through resampling.

To obtain the eigenvalues, let **R** be the observed $p \times p$ sample correlation matrix. Since **R** is symmetric and positive definite, then orthogonal decomposition will give the eigenvalues and eigenvectors, and these eigenvalues are guaranteed to be real and positive. That is,

$$\mathbf{R}=\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{T}} \text{ where } \boldsymbol{\Lambda}=diag(\lambda_1,\lambda_2,\ldots\lambda_p) \quad \text{and} \quad \mathbf{Q}=\left\lfloor \mathbf{e}_1 \vdots \mathbf{e}_2 \vdots \cdots \vdots \mathbf{e}_p \right\rfloor \text{ where } \lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_p \ge 0$$

where **Q** is a square $p \times p$ orthogonal matrix and **Λ** is the diagonal matrix whose diagonal elements are the corresponding eigenvalues of **R**. The columns of the matrix $\mathbf{Q}\,\mathbf{Q}^T$ are the eigenvectors of **R**.

False Discovery Rate Parallel Analysis Procedure

1. For each of the n observations (rows) in the sample, permute the p variable labels among the variable values (columns). The permutations should be independent for each observation. The result is a permutation sample. Obtain the eigenvalues from the sample correlation matrix.
2. Repeat the previous step one thousand times. The result will be one thousand sets of eigenvalues from each of the one thousand permutation samples. Save the eigenvalues, but not the permutation samples or their correlation matrices.
3. Choose an initial eigenvalue cutoff e_0; any positive real number will suffice.
4. Count the number of eigenvalues from the observed sample that are greater than e_0; this is called the observed discovery, denoted x_{OD}. For each of the one thousand permutation samples, count the number of eigenvalues greater than e_0 and average these counts together; this is called the false discovery, denoted x_{FD}. Calculate the false discovery rate: $FDR = x_{FD} / x_{OD}$.
5. Using a bisection search algorithm, iteratively repeat step (4) to determine the eigenvalue cutoff e_{FD} corresponding to a false discovery rate of approximately 10%. Count the number of eigenvalues from the observed sample correlation matrix which are greater then e_{FD}, this is the number of factors to retain in factor analysis.

Factor Analysis of The Assistments Survey Data

In this section, we analyze the factor structure of the two surveys administered during the Assistments experiment. We employed the methods of parallel analysis and maximum likelihood factor analysis discussed in the previous two sections.

To determine the number of underlying factors in the SUMS and grit surveys, we performed independent parallel analyses, each with false discovery rates of ten percent. The results are shown in Table 1. Eigenvalue cutoff values and their corresponding number of factors are shown for several false discovery rates. For both surveys, a ten percent false discovery rate suggests one underlying factor.

Table 1: Results of False Discovery Rate (FDR) Parallel Analysis on Survey Data

	Grit Survey		SUMS Survey	
FDR	Cutoff e_0	Factors m	Cutoff e_0	Factors m
0.50	2.43	1	2.79	1
0.25	2.47	1	2.82	1
0.10	2.51	1	2.85	1
0.05	2.53	1	2.87	1
0.01	2.57	1	2.92	1

Having determined the number of factors in each of the surveys, we performed maximum likelihood factor analysis to estimate the factor loadings. The results are presented in Table 2, which contains the estimated factor loadings, specific errors, and proportion of variation explained by each model.

Table 2: Factor Analysis of Survey Data

Item	λ	ψ	Item	λ	ψ
GRIT02	**0.568**	0.678	SUMS01	**0.361**	0.869
GRIT03	**0.334**	0.888	SUMS03	**0.597**	0.644
GRIT04	**0.653**	0.574	SUMS04	**0.371**	0.863
GRIT05	**0.645**	0.584	SUMS05	**0.480**	0.770
GRIT06	-0.154	0.976	SUMS07	**0.493**	0.757
GRIT07	0.251	0.937	SUMS10	**0.547**	0.701
GRIT08	-0.240	0.942	SUMS12	**0.439**	0.807
GRIT09	-0.141	0.980	SUMS13	**0.501**	0.749
GRIT10	**0.593**	0.648	SUMS15	**0.346**	0.880
GRIT11	**0.605**	0.634	SUMS16	**0.550**	0.698
GRIT12	-0.200	0.960	SUMS18	**0.465**	0.784
GRIT13	**0.489**	0.761	SUMS19		0.999
GRIT14	**0.771**	0.406			
Variation Explained		0.233	Variation Explained		0.207

Note: Blank spaces are factor loadings less than 0.10, which are not outputted by the factor analysis procedure in R. Loadings greater than 0.30 are shown in bold. The "Variation Explained" row denotes the proportion of the variance in the survey data accounted for by the factor model.

SECTION 4: STRUCTURAL EQUATION MODELS

Structural equation models (SEM) combine factor analysis and multivariate regression models to estimate associations between observable variables and unobservable (latent) variables. They are commonly used in social sciences, since they allow the analysis of unobserved latent variables, such as intelligence and perseverance. Multiple linear regression and factor analysis are special cases of structural equations models.

In this section we review and interpret structural equation models. We performed a simulation study to evaluate the correctness of estimation using SEM and compared these estimates with those from traditional methods of regression analysis and factor analysis. Finally, we analyzed the Assistments data using structural equation models for bivariate response data.

Model and Assumptions

Structural equation models have two components: the measurement model, which relates the observed variables to latent variables; and the structural model which relates the latent variables among each other. The structural component is not necessary, and was not used in this project.

The structural equation measurement model has the following form:

$$\mathbf{y}_i = \boldsymbol{\Lambda}_y \boldsymbol{\eta}_i + \boldsymbol{\varepsilon}_i \qquad \boldsymbol{\varepsilon}_i \sim N_p(\mathbf{0}, \boldsymbol{\Sigma}_y)$$
$$\mathbf{x}_i = \boldsymbol{\Lambda}_x \boldsymbol{\xi}_i + \boldsymbol{\delta}_i \qquad \boldsymbol{\delta}_i \sim N_q(\mathbf{0}, \boldsymbol{\Sigma}_x)$$

with the following assumptions:

$$E(\boldsymbol{\eta}) = 0, \qquad E(\boldsymbol{\xi}) = 0, \qquad E(\boldsymbol{\varepsilon}) = 0, \qquad E(\boldsymbol{\delta}) = 0,$$

$$\mathrm{Var}(\boldsymbol{\eta}) = \mathbf{I}, \qquad \mathrm{Var}(\boldsymbol{\xi}) = \mathbf{I},$$

$$\mathrm{Cov}(\boldsymbol{\varepsilon}, \boldsymbol{\eta}) = 0, \qquad \mathrm{Cov}(\boldsymbol{\varepsilon}, \boldsymbol{\xi}) = 0, \qquad \mathrm{Cov}(\boldsymbol{\varepsilon}, \boldsymbol{\delta}) = 0, \qquad \mathrm{Cov}(\boldsymbol{\delta}, \boldsymbol{\xi}) = 0, \qquad \mathrm{Cov}(\boldsymbol{\delta}, \boldsymbol{\eta}) = 0$$

The observed vectors $\mathbf{y}_i$ and $\mathbf{x}_i$ are multivariate normal with p and q variables, respectively. In the model, $\boldsymbol{\Lambda}_y$ and $\boldsymbol{\Lambda}_x$ are matrices of factor loadings. The vectors $\boldsymbol{\eta}_i$ and $\boldsymbol{\xi}_i$ are latent variables, vectors $\boldsymbol{\varepsilon}_i$ and $\boldsymbol{\delta}_i$ are normally distributed errors or residuals, and $\boldsymbol{\Sigma}_y$ and $\boldsymbol{\Sigma}_x$ are covariance matrices – typically chosen to be diagonal.

Existing structural equation modeling software packages such as R, SAS, and LISREL require the covariance matrix of the observed variables and the number of observations in the sample as inputs. The software packages use many methods for estimating the unknown parameters in SEM, such as maximum likelihood, unweighted least squares, and two-stage least squares. These methods essentially estimate the unknown parameters such that the implied covariance matrix is close to the sample covariance matrix. For a more thorough discussion of the estimation of parameters in SEM, see (Bollen, 1989, pp. 104-123).

To use SEM estimation, we must assume our data are multivariate normal. In the case of the Assistments data, we know these data are not normally distributed. The survey response data are ordinal (Likert scale), as are the pre-test scores and post-test scores. Polychoric correlation matrices could be used instead of regular correlation matrices; see (Jöreskog, 2005) for details on polychoric correlations.

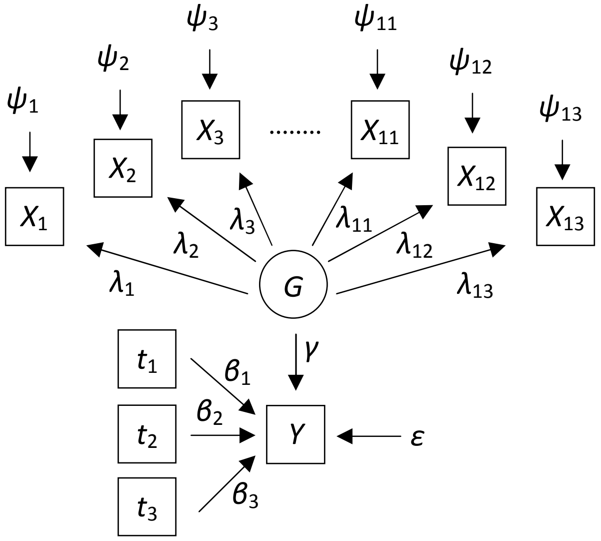
Interpretation

Consider the following structural equation model, based on the Assistments data:

$$\mathbf{X}_i = \mathbf{L}G_i + \boldsymbol{\psi}_i$$
$$Y = t_{i,1}\beta_1 + t_{i,2}\beta_2 + t_{i,3}\beta_3 + \gamma G_i + \varepsilon_i$$

Let $\mathbf{X}$ be the vector of survey responses from the grit survey, and G is the latent unobservable factor representing students' grit. Let t_1, t_2, and t_3 be treatment indicator variables with corresponding treatment effects β_1, β_2, and β_3. Finally, let γ be the effect of the latent grit factor. The first equation is a factor model relating survey responses to an unobservable factor. The second equation is a regression model measuring the treatment effects, and the effect of the latent grit factor. The path diagram for this structural equation model is illustrated in Figure 5.

Figure 5: Path Diagram of a Structural Equation Model



The path diagram of the structural equation model in Figure 5 contains two sections. The upper section represents a factor model relating grit survey responses X_1, …, X_{13} to G, the grit factor. The lower section represents a multiple linear regression model with response variable Y, treatment effects β_1, β_2, β_3, grit effect γ, and error term ε.

Simulation Study

We used simulation study to determine whether statistical procedures perform as expected. Of particular interest in this project is whether the structural equation procedure “sem()” available in the R statistical software package can correctly estimate the unknown model parameters.

We simulated two structural equation models: a simple model, and a more complex model with interaction and treatment effects, which will later be applied to the Assistments data. We compared the performance of SEM estimation with that of a **two-stage** method, which applies factor analysis to estimate the latent factors, then sequentially applies multiple linear regression analysis. The validity of confidence statements in confidence intervals estimated by two methods are compared.

Simulation Study of a Simple Structural Equation Model

Consider the following simple structural equation model with latent factor G, intercept β, and slope γ. The model contains no interaction terms and no treatment effects.

$$\mathbf{X}_i = \mathbf{L} G_i + \boldsymbol{\delta}_i \quad \text{and} \quad Y_i = \beta + \gamma G_i + \varepsilon_i$$

The Simulation Study Procedure

1. Select β, γ, the factor loadings $\mathbf{L}$, and the specific errors $\boldsymbol{\psi}$.

2. Generate G_i and ε_i from the standard normal distribution and generate δ_i from the multivariate normal distribution with diagonal covariance matrix, where the diagonal elements are equal to $\boldsymbol{\psi}$. For a model with treatments, assign observations to treatments with n = 3000 observations per treatment.
3. Calculate X_i and Y_i.
4. Estimate β and γ using the two-stage method of factor analysis followed by regression analysis.
5. Estimate β and γ using the one-stage structural equation model procedure.
6. Repeat steps (2) - (5) one thousand times. For each replicate, record the coefficient estimates and their standard errors. Calculate 95% confidence intervals and check whether they contain their actual parameter values, chosen in step (1).

The procedure will produce one thousand simulated samples. For each sample, the coefficients and their standard errors are estimated using the SEM procedure and the two-stage estimation method. For each coefficient estimate, we determine whether the Estimated 95% confidence interval contains the true parameter value. The **capture rate** is the portion of these estimated 95% confidence intervals which contain the actual parameter value; this is the quantity we are interested in.

The results of the simulation study are presented in Table 3. For both estimation methods, the mean and median of the one thousand parameter estimates are exactly the same as the true parameter values. The capture rate of SEM estimation was exactly 95% overall, meaning that 95% of the estimated confidence intervals contained their true parameter values. The capture rate of the two-stage estimation method was 72%, much lower than the SEM method. The cost of applying two independent iterations of maximum likelihood in the two-stage method is apparent in the low capture rate. Part of the reason we achieved a 95% capture rate is because the sample size was large (n = 3000). With smaller sample size, the capture rate will decrease. For example, with 30 observations, we observed a 93.1% capture rate.

Table 3: Simulation Study Parameter Estimates for the Simple Model

		SEM Estimation			Two-Stage Estimation		
Parameter	Actual	Mean	Median	Capture Rate	Mean	Median	Capture Rate
β	1.50	1.50	1.50	95.0%	1.50	1.50	64.2%
γ	2.50	2.50	2.50	94.9%	2.50	2.50	79.7%
Overall				95.0%			72.0%

Simulation Study of a Complex Structural Equation Model

Consider the following structural equation model; this is the **ideal** model we would like to apply to the Assistments data:

$$Y_{i,k} = \beta_k + \gamma_k G_i + \varepsilon_i$$
$$X_i = L G_i + \delta_i$$
$$k = 1,\ 2,\ 3$$

In the ideal model, the response variable Y is the overall test score gain, X is the vector of grit survey responses, and β_k are intercepts for students under the k^{th} learning condition. The grit factor is denoted G with grit-treatment interaction effects γ_k. The regression model error terms are denoted by ε, and L is the usual matrix of factor loadings with error vector δ.

The **ideal** model above is the model we want to estimate. However, structural equation modeling software available today has a major disadvantage: the software cannot directly estimate interaction effects between an observable variable and an unobservable variable. In order to estimate the interaction effects we must use a **compromise** model instead:

$$Y_{i,k} = \beta_k + \gamma_k G_{i,k} + \varepsilon_i$$
$$X_i = L_k G_{i,k} + \delta_i$$
$$k = 1,\ 2,\ 3$$

In the ideal model, the grit factor is estimated once for all three treatments; in the compromise model, each treatment group has its own estimated grit factor, with different factor loading matrices. In the compromise model, information about the grit factor is not shared across treatment groups. We partition the sample into three sub-samples, such that all of the observations from treatment *k* are in sub-sample *k*. Then we fit sub-model *k* to sub-sample *k*. The simulation procedure for estimation is effectively the same as the procedure used for the simple model. Instead of two parameters of interest (β, γ), we have six (β_1, β_2, β_3, γ_1, γ_2, γ_3). In our analysis, the two-stage estimation method is applied directly to the ideal model and the SEM estimation method is applied to the compromise model.

Table 4: Simulation Study Parameter Estimates for the Full Model

		SEM Estimation			Two-Stage Estimation		
Parameter	Actual	Mean	Median	Capture Rate	Mean	Median	Capture Rate
β_1	1.50	1.50	1.50	94.5%	1.50	1.50	73.6%
β_2	2.50	2.50	2.50	94.5%	2.50	2.50	87.1%
β_3	3.50	3.50	3.50	94.4%	3.50	3.50	96.3%
γ_1	3.00	3.00	3.00	94.7%	3.00	3.00	84.8%
γ_2	2.00	2.00	2.00	94.6%	2.00	2.00	92.7%
γ_3	1.00	1.00	1.00	93.8%	1.00	1.00	96.7%
Overall				94.4%			88.5%

The results of the simulation study are presented in Table 4. For both estimation methods, the mean and median parameter estimates are same as the true parameter values. The SEM estimation capture rate was 94.4% overall. The capture rate is lower than 95% because we used the compromise model. The two-stage estimation capture rate was 88.5%, still quite lower than the SEM method, even though the ideal model was used for the two-stage method. There is another notable observation from Table 4. The capture rates for SEM estimation are consistently about 95% for all six parameters. In contrast, the two-stage estimation capture rates varied wildly across the six parameters, ranging from 73.6% to 96.7%. This inconsistency in estimation is likely a consequence of applying two iterations of maximum likelihood estimation.

Preliminary Analysis of Assistments Data

In this section the Assistments dataset is presented and the statistical tests we conducted are presented; these analyses focused on testing treatment randomization, testing for within-treatment gains, and testing for treatment effect. Throughout this section we require an α = 0.05 significance level.

The data in Table 5 is a fabricated sample of the Assistments dataset. Each row represents one student. For each student we have the grade level, treatment assignment, scores for the ramp experiment and inquiry pre-tests and post-tests, and responses to the grit and SUMS surveys.

Table 5: The Assistments Data, A Fabricated Example

Student	Grade	Treatment	Ramp Pre/Post		Inquiry Pre/Post		Grit Survey Responses	SUMS Survey Responses
1	7	1	2	4	1	3	{1,5,3,2,5,3,5,3,2,5,4,5,4}	{5,1,4,1,1,3,2,1,3,2,3,3}
2	7	2	4	2	3	2	{5,3,4,3,1,1,1,1,3,3,2,4,4}	{1,1,2,5,4,2,3,1,2,4,4,1}
3	7	3	1	2	3	3	{3,5,4,3,3,2,4,5,4,2,2,3,2}	{4,1,3,1,5,2,3,4,2,2,2,2}
4	8	1	1	2	3	1	{4,1,5,4,2,2,2,1,4,4,1,4,5}	{3,1,4,2,3,4,2,4,2,4,3,3}
5	8	2	3	4	1	1	{4,1,1,5,4,4,3,4,5,4,2,2,4}	{1,4,3,2,2,2,1,3,3,1,1,5}
6	8	3	2	2	1	3	{4,2,5,3,2,4,1,5,3,2,5,1,1}	{3,4,4,3,4,5,1,2,1,3,1,3}

Each student was randomly assigned a treatment group *after* completing the ramp experiment pre-test and the inquiry pre-test. The assignment to treatment groups was random, so there should be no difference in mean pre-test scores across treatment groups. To test this randomization of treatment groups, we performed an analysis of variance (ANOVA) with the pre-test scores as the response variables, and the treatments as the group variables. The ANOVA results do not provide sufficient evidence to suggest a difference in mean pre-test scores across treatment groups. The p-value for the ramp experiment pre-test was 0.4148, and the p-value for the inquiry pre-test was 0.1304. Note that fewer students completed the inquiry pre-test than the ramp pre-test, so the degrees of freedom are different.

We then explored whether there was a difference between pre-test scores and post-test scores; in other words, we checked whether the students learned from the ramp microworld, regardless of the learning condition (treatment group). To test whether there were gains in test scores, we performed t-tests for each treatment group; the results are summarized in Table 6 and Table 7. From these tables, we conclude that there were significant gains in experiment test scores across all three treatment groups. However, there was not sufficient evidence to suggest that the inquiry pre-test scores were different from the post-test scores.

Table 6: T-Tests for Within-Treatment Gains on Ramp Experiment Tests

Treatment	DF	Mean	95% C.I.	P-Value
1	56	0.632	(0.181, 1.082)	0.0068
2	56	0.316	(0.036, 0.596)	0.0277
3	52	0.642	(0.326, 0.957)	0.0002

Note: The treatment groups were coded as follows: 1 = Direct, 2 = Discovery, and 3 = Direct + Reify. Refer to the experimental procedure for treatment group descriptions.

Table 7: T-Tests for Within-Treatment Gains on Inquiry Tests

Treatment	DF	Mean	95% C.I.	P-Value
1	49	0.180	(-0.117, 0.477)	0.2285
2	43	0.250	(-0.079, 0.579)	0.1323
3	46	0.191	(-0.164, 0.547)	0.2833

Analysis of Assistments Data with SEM

The structural equation model used to analyze the Assistments data was based on the compromise model from the previous section. We modeled inquiry test score gains, Y_{I}, and ramp experiment test score gains, Y_{R}, as the bivariate response variable $\mathbf{Y} = (Y_{\mathrm{I}}, Y_{\mathrm{R}})$. In the Grit Model, gain scores are modeled as a function of treatment effects and the effect of a latent grit factor G. In the SUMS Model, gain scores are modeled as a function of treatment effects and the effect of a latent SUMS factor S.

The Grit Model

$$\begin{bmatrix} Y_{I,i,k} \\ Y_{R,i,k} \end{bmatrix} = \begin{bmatrix} \beta_{I,k} \\ \beta_{R,k} \end{bmatrix} + \begin{bmatrix} \gamma_{I,k} \\ \gamma_{R,k} \end{bmatrix} G_{i,k} + \begin{bmatrix} \varepsilon_{I,i} \\ \varepsilon_{R,i} \end{bmatrix}$$

$$\mathbf{X}_i = \mathbf{L}_k\, G_{i,k} + \boldsymbol{\delta}_i$$

$$k = 1,\ 2,\ 3$$

The SUMS Model

$$\begin{bmatrix} Y_{I,i,k} \\ Y_{R,i,k} \end{bmatrix} = \begin{bmatrix} \beta_{I,k} \\ \beta_{R,k} \end{bmatrix} + \begin{bmatrix} \gamma_{I,k} \\ \gamma_{R,k} \end{bmatrix} S_{i,k} + \begin{bmatrix} \varepsilon_{I,i} \\ \varepsilon_{R,i} \end{bmatrix}$$

$$\mathbf{X}_i = \mathbf{L}_k\, G_{i,k} + \boldsymbol{\delta}_i$$

$$k = 1,\ 2,\ 3$$

The models are essentially bivariate versions of the compromise model from the previous section with terms analogous to those of the compromise model. The effects coefficients are vectors $(\beta_{I,k}, \beta_{R,k})^{\mathrm{T}}$ and $(\gamma_{I,k}, \gamma_{R,k})^{\mathrm{T}}$, with ($k$ = 1, 2, 3). For example, the effect of treatment group k = 1 on inquiry test score gain is denoted $\beta_{I,1}$ in the SEM model.

The structural equation model coefficient estimates of interest for both models are presented in Table 8. The β's represent the average test score gain for a student with zero grit which has no practical meaning. The difference in coefficients ($\beta_{R,2}$ - $\beta_{R,1}$) represents the difference in test score gains between conditions 2 and 1 for students with the same grit. The difference ($\gamma_{R,2}$ - $\gamma_{R,1}$) represents the additional test score gain for one unit increase of grit. The coefficient differences and their associated p-values are presented in Table 9. There were no significant differences, meaning no treatment conditions resulted in significantly higher (or lower) test score gains than another treatment condition. With a larger sample size, and presumably smaller standard errors, some meaningful differences might emerge.

Table 8: SEM Parameter Estimates

	SUMS Model		GRIT Model	
Parameter	Estimate	Standard Error	Estimate	Standard Error
$\gamma_{I,1}$	-1.2893	1.0519	0.0348	0.7759
$\beta_{I,1}$	1.3666	1.0396	0.1567	0.7701
$\gamma_{R,1}$	-0.1908	1.6989	-1.0346	1.2895
$\beta_{R,1}$	0.7244	1.6948	1.5969	1.2752
$\gamma_{I,2}$	-2.8867	2.1714	0.5413	0.8520
$\beta_{I,2}$	3.1263	2.1350	-0.3421	0.8434
$\gamma_{R,2}$	2.0088	2.3953	-1.5483	0.8593
$\beta_{R,2}$	-1.6734	2.3761	1.9835	0.8323
$\gamma_{I,3}$	-1.8893	1.5087	0.6178	0.8200
$\beta_{I,3}$	2.2030	1.4891	-0.2730	0.8120
$\gamma_{R,3}$	0.4804	1.3821	-1.1112	0.7709
$\beta_{R,3}$	0.2190	1.3783	1.7256	0.7561

Table 9: SEM Coefficient Differences

	SUMS Model				GRIT Model			
Difference	Estimate	Standard Error	Z-Value	P-Value	Estimate	Standard Error	Z-Value	P-Value
$\gamma_{I,2} - \gamma_{I,1}$	-1.5974	2.4128	-0.6621	0.5079	0.5064	1.1524	0.4395	0.6603
$\gamma_{I,3} - \gamma_{I,1}$	-0.5999	1.8392	-0.3262	0.7443	0.5830	1.1289	0.5164	0.6055
$\gamma_{I,3} - \gamma_{I,2}$	0.9975	2.6441	0.3772	0.7060	0.0766	1.1825	0.0647	0.9484
$\beta_{I,2} - \beta_{I,1}$	1.7597	2.3747	0.7410	0.4587	-0.4989	1.1421	-0.4368	0.6623
$\beta_{I,3} - \beta_{I,1}$	0.8364	1.8161	0.4605	0.6451	-0.4298	1.1190	-0.3840	0.7009
$\beta_{I,3} - \beta_{I,2}$	-0.9233	2.6030	-0.3547	0.7228	0.0691	1.1707	0.0590	0.9529
$\gamma_{R,2} - \gamma_{R,1}$	2.1996	2.9366	0.7490	0.4538	-0.5137	1.5495	-0.3315	0.7402
$\gamma_{R,3} - \gamma_{R,1}$	0.6712	2.1901	0.3065	0.7593	-0.0766	1.5023	-0.0510	0.9593
$\gamma_{R,3} - \gamma_{R,2}$	-1.5285	2.7655	-0.5527	0.5805	0.4371	1.1544	0.3786	0.7050
$\beta_{R,2} - \beta_{R,1}$	-2.3977	2.9186	-0.8215	0.4113	0.3866	1.5227	0.2539	0.7996
$\beta_{R,3} - \beta_{R,1}$	-0.5053	2.1845	-0.2313	0.8171	0.1287	1.4825	0.0868	0.9308
$\beta_{R,3} - \beta_{R,2}$	1.8924	2.7469	0.6889	0.4909	-0.2579	1.1244	-0.2294	0.8186

In summary, using structural equation models we tested the effects of the three learning conditions on students' scientific inquiry test score gains. Each model controlled for an unobservable latent trait hidden within survey response data, and also measured the interaction between the learning conditions and the unobservable trait. The key results from this analysis in Table 9 indicate no significant test score gains from treatment effects or interaction effects; in other words, no learning condition was significantly better or worse than any other learning condition, when controlling for the latent traits.

REFERENCES

1. Abdi, Herve
   Factor Rotations in Factor Analyses
   http://www.utdallas.edu/~herve/Abdi-rotations-pretty.pdf

2. Benjamini, Y., & Hochberg, Y. (1995)
   Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing
   Journal of the Royal Statistical Society, Series B, 57(1): 289-300

3. Bollen, Kenneth A.
   Structural Equations with Latent Variables (pp. 32-34)
   New York: John Wiley, 1989

4. Duckworth, A. L., Peterson, C., Matthews, M. D., & Kelly, D. R. (2007)
   Grit: Perseverance and passion for long-term goals
   Journal of Personality and Social Psychology, 92(6), 1087-1101

5. Dunson, David B. (2007)
   Bayesian Methods for Latent Trait Modeling of Longitudinal Data
   Statistical Methods in Medical Research 16, 399-415.

6. Fox, John (2006)
   Structural Equation Models: Appendix to an R and S-PLUS Companion to Applied Regression
   http://cran.r-project.org/doc/contrib/Fox-Companion/appendix-sems.pdf

7. Fox, John (2008).
   sem: Structural Equation Models. R package version 0.9-13.
   http://www.r-project.org, http://www.socsci.mcmaster.ca/jfox/

8. Gobert, Janice; et al.
   Annual Report for Year 1 for AMI: ASSISTments Meets Inquiry (NSF-DRL# 0733286)
   (unpublished)

9. Johnson, R.A., and D.W. Wichern
   Applied Multivariate Statistical Analysis
   Pearson Prentice-Hall, 2007 Sixth Edition, pp. 481-497

10. Jöreskog, Karl (2005)
    Structural Equation Modeling with Ordinal Variables using LISREL
    http://www.ssicentral.com/techdocs/ordinal.pdf

11. Kline, Rex B.
    Principles and Practice of Structural Equation Modeling, Second Edition (pp. 66-69)
    New York: The Guilford Press, 2005

12. National Committee on Science Education Standards and Assessment (1996).
    National Science Education Standards (NSES)
    Washington, DC: National Academy Press, http://www.nap.edu

13. Pohlmann, John T.
    Use and Interpretation of Factor Analysis in the Journal of Educational Research: 1992-2002.
    Journal of Educational Research, v98 n1 p14 Sep 2004

14. R Development Core Team (2008).
    R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

15. Richardson, Jacqui; (2008)
    Science ASSISTments: Tutoring Inquiry Skills in Middle School Students
    Interactive Qualifying Project Report. Worcester Polytechnic Institute. May 2008. Unpublished.

16. Treagust, David F.; Chittleborough, Gail; Mamiala, Thapelo L.
    Students' Understanding of the Role of Scientific Models in Learning Science
    International Journal of Science Education, v24 n4 p357-68 Apr 2002

APPENDIX A: SOURCE CODE

```
# LOAD THE ASSISTMENTS DATA
# -------------------------

ds.data <- read.table("C:\\ASSISTMENTS.TXT", header = TRUE, sep="\t")

ds.grit <- na.omit(ds.data[,17:29]) # Select the GRIT survey items
ds.sums <- na.omit(ds.data[,05:16]) # Select the SUMS survey items


# PRELIMINARY ANALYSIS OF ASSISTMENTS DATA
# ----------------------------------------

# Treatment Randomization Tests
# -----------------------------

ds.lm <- lm( RAMP_PRE ~ factor(TREATMENT), data=ds.data)
anova(ds.lm)

ds.lm <- lm( INQ_PRE  ~ factor(TREATMENT), data=ds.data)
anova(ds.lm)

# T-Tests for Within-Treatment Gains
# ----------------------------------

t.test( subset(ds.data, TREATMENT==1)$RAMP_GAIN )   # Test for ramp experiment gains
t.test( subset(ds.data, TREATMENT==2)$RAMP_GAIN )   #
t.test( subset(ds.data, TREATMENT==3)$RAMP_GAIN )   #

t.test( subset(ds.data, TREATMENT==1)$INQ_GAIN )    # Test for inquiry gains
t.test( subset(ds.data, TREATMENT==2)$INQ_GAIN )    #
t.test( subset(ds.data, TREATMENT==3)$INQ_GAIN )    #

# ANOVA for Treatment Effects on Post-Test Scores
# -----------------------------------------------

ds.lm <- lm( RAMP_POST ~ factor(TREATMENT) + RAMP_PRE, data=ds.data)
anova(ds.lm)
summary(ds.lm)

ds.lm <- lm(  INQ_POST ~ factor(TREATMENT) + INQ_PRE , data=ds.data)
anova(ds.lm)
summary(ds.lm)

# DETERMINING THE NUMBER OF FACTORS WITH PARALLEL ANALYSIS
# --------------------------------------------------------
# The following program will perform false-discovery-rate parallel analysis on the GRIT or SUMS surveys. The
# procedure is used to estimate the number of factors to estimate in factor analysis (or SEM).

 ds <- ds.grit          # Change from "ds.grit" to "ds.sums" to analyze the SUMS data
 rs <- ds               # Resampled Dataset via Random Permutation
  n <- nrow(ds)         # Number of Obserations
  p <- ncol(ds)         # Number of Survey Items
  r <- 10               # Replication Count
 ev <- matrix( ,r, p)   # Eigenvalues for each Replicate

# Resampling Responses
----------------------
# Each observation in the original dataset is a vector of p survey responses. We randomly permute the survey
# responses within each observation to generate a new new dataset. Then we calculate the eigenvalues of the new
# resampled dataset. We repeat this procedure 1,000 times, and the result is a 1000 x p matrix of eigenvalues.

for( k in 1:r)                     # For each replicate
{ for( i in 1:n )                  # For each observation in the dataset
  { rs[i,] <- sample(ds[i,]) }     # Randomly permute the survey responses within observation
  ev[k,] <- eigen(cor(rs))$values  # Compute the eigenvalues of the resampled dataset
}

# Bisection Search
# ----------------
# The following algorithm will use bisection search to determine the observed discovery OD that corresponds to a
# false discovery rate target of FD/OD = 0.10.

e_min = 0               # Lower bound for bisection search
e_max = 5               # Upper bound for bisection search
```

```
e <- (e_min+e_max)/2    # Bisection (initial guess)
td <- 0.10              # Target FD/OD Ratio

for( i in 1:20)         # Max of 20-Iterations
{
  fd <- array(0, r)                   # Empty array of eigenvalue counts
  od <- sum(eigen(cor(ds))$values>e)
  for( k in 1:r)                      # For each replicate
  {
    fd[k] <- sum(ev[k,] > e )         # Count number of eigenvalue greater than cutoff
  }

  if( od == 0 ) e_max <- e

  if( od > 0 )
  {
     if( mean(fd)/od > td ) e_min <- e  # Determine new bisection window
     if( mean(fd)/od < td ) e_max <- e
  }
  e <- (e_min + e_max)/2                # Bisection
  if ( e_max - e_min < 0.001 ) break    # Break if convergence is achieved
}

e; od; mean(fd); mean(fd)/od;                      # Statistics of interest
ds.grit.factors <- sum(eigen(cor(ds))$values>e)    # Resulting number of factors to retain change this
                                                   # to “ds.sums.factors” when analyzing the SUMS survey
# FACTOR ANALYSIS
-----------------
This section will perform factor analysis on the GRIT and SUMS surveys. It assumes that the number of factors
has been pre-determined using parallel analysis (previous section), and that the number of factors to retain in
stored in the variables “ds.grit.factors” and “ds.sums.factors”

ds.grit.fa <- factanal(ds.grit, factors=ds.grit.factors) # GRIT Factor Analysis
ds.sums.fa <- factanal(ds.sums, factors=ds.grit.factors) # SUMS Factor Analysis

# SIMULATION STUDY OF STRUCTURAL EQUATION MODELS
# ----------------------------------------------
#
#  Symbol   Description in Study
#  ---------------------------------------------------------
#    Y      Test Score Gains (Post Test – Pre Test)
#    X      Survey Response Data, Each of the 13 Survey Items = {1,2,3,4,5} or NA
#    b      Effect of Observed Covariate   “beta”
#    g      Effect of Latent Factor        “gamma”
#    G      Grit (Latent Factor)
#    t1     Treatment Group 1, Indicator Variable (0,1)
#    t2     Treatment Group 2, Indicator Variable (0,1)
#    t3     Treatment Group 3, Indicator Variable (0,1)
#    e      Regression Analysis Observation-Wise Errors (epsilon)
#    d      Factor Analysis Observation-Wise Errors (delta)
#    L      Factor Analysis Loadings Matrix
#    psi    Factor Analysis Specific Errors
#    n      Number of Observations in the Sample
#    i      Index for Observations in the Sample, (i = 1 to n)
#    u      Unit Vector (Intercept)
#    r      Index for Replicates

# Methodology
#
# A. Select b, g, L, and psi.
# B. Generate G[i], e[i] from normal distribution, d[i] from multivariate normal distribution.
# C. Calculate X[i] and Y[i].
# D. Estimate model using “two-stage” method. Save coefficient estimates and standard errors
# E. Estimate model using “one-stage” method. Save coefficient estimates and standard errors
# F. Repeat (B-E) one thousand times.

# PART ONE: THE SIMPLE MODEL, Y = b + g*G + e
#

# Step 1: Load Packages, Specify Path Diagram
# -------------------------------------------
# The first step is to load the required R packages and specify the path diagram for our model. The “sem”
# library is needed to perform structural equation modeling, and the “MASS” library is needed to sample from
# the multivariate normal distribution.
#

library(sem)                # Library: Structural Equation Modeling
library(MASS)               # Library: Modern Applied Statistics
```

```
ds.model <- specify.model()
G   ->  x01, lamda01, NA
G   ->  x02, lamda02, NA
G   ->  x03, lamda03, NA
G   ->  x04, lamda04, NA
G   ->  x05, lamda05, NA
G   ->  x06, lamda06, NA
G   ->  x07, lamda07, NA
G   ->  x08, lamda08, NA
G   ->  x09, lamda09, NA
G   ->  x10, lamda10, NA
G   ->  x11, lamda11, NA
G   ->  x12, lamda12, NA
G   ->  x13, lamda13, NA
y   <-    u, intercept, NA
y   <-    G, lamda_G, NA
G   <->   G, NA, 1
y   <->   y, e, NA
x01 <-> x01, delta01, NA
x02 <-> x02, delta02, NA
x03 <-> x03, delta03, NA
x04 <-> x04, delta04, NA
x05 <-> x05, delta05, NA
x06 <-> x06, delta06, NA
x07 <-> x07, delta07, NA
x08 <-> x08, delta08, NA
x09 <-> x09, delta09, NA
x10 <-> x10, delta10, NA
x11 <-> x11, delta11, NA
x12 <-> x12, delta12, NA
x13 <-> x13, delta13, NA

# Step 2: Initialize Replication Counter, or Re-Enter Interrupted Code
# --------------------------------------------------------------------
# The goal is to accumulate results from one thousand replicates. After each replicate finishes, the results are
# appended to a text file. If R crashes before finishing all one thousand replicates, we import the text file to
# determine the last replicate that finished. Then we continue where we left off.

r<-1                         # Initialize replicate index
options(warn=-1)              # Suppress warning messages

# Look for the results file. If the results file exists, then figure out the correct value of r. Otherwise the
# file does not exist, and r is left unchanged, so r=1.

try(temp <-read.table("c:\\SIMPLE_1STAGE.TXT", header = FALSE, sep="\t"))
try(r <- max(temp[,1]) + 1)

while (r <= 10  )                           # Loop through 1000 replicate
{

  # Step 3: Specify Regression Effects
  # ----------------------------------

  n <-30                                    # Number of Observations
  u <- rep(1,n)                             #
  b <- rbind(1.5, 2.5)                      # Beta (b) = 1.5 and Gamma (g) = 2.5
  rownames(b) = c("Intercept","Slope")      #

  # Step 4: Factor Loadings, Specific Errors, Random Errors
  # -------------------------------------------------------
  # (1) Generate GRIT Factor ~ N(0,1)
  # (2) Specify L and Psi from previous research.
  # (3) Generate errors: d(elta) and e(psilon)

  G<-rnorm(n)

  L   <-rbind(0.501, 0.599, 0.591, 0.756, -0.531, 0.527, -0.643, -0.466, 0.686, 0.477, -0.531, 0.740, 0.622)
  psi <-    c(0.749, 0.641, 0.650, 0.429,  0.718, 0.723,  0.587,  0.783, 0.529, 0.772,  0.718, 0.453, 0.613)

  d <-mvrnorm(n,rep(0,13),diag(psi))
  e <- rnorm(n,0,1)

  # Step 5: Calculate X, Y. Build Final Dataset
  # -------------------------------------------

  X<-matrix(rep(0,13*n),n,13)
  Y<-rep(0,n)
  for(i in 1:n)
  {
    X[i,] <- L %*% G[i] + d[i,]
    Y[i]  <- b[1] + b[2]*G[i] + e[i]
  }
  ds <- data.frame(cbind(Y,u,X))
  names(ds) = cbind("y","u","x01","x02","x03","x04","x05","x06","x07","x08","x09","x10","x11","x12","x13")

  # Step 6: Two-Stage Estimation using Factor Analysis and Regression
  # -------------------------------------------------------------------
  # (1) Factor analysis to estimate latent factor G
  # (2) Multiple linear regression analysis to estimate b and g

  ds.fa <- factanal(x = ds[,3:15], factors=1, scores="regression")   # Factor Analysis
  ds <- data.frame(cbind(ds, ds.fa$scores))                          # Get Factor Scores
  colnames(ds)[16]="g"                                               #

  ds.lm <- lm( y ~ g, data=ds )                                      # Linear Regression
  ds.lm.b <- summary(ds.lm)$coefficients[1:2,1:2]                    # Get Coefficients, Standard Errors

  # Step 7: One-Stage Estimation using Structural Equation Modeling
  # -----------------------------------------------------------------
  # When the number of observations is small, the sem() procedure sometimes refuses to estimate our paramters.
  # This behavior is not mentioned in the sem() documentation. The sem() procedure is executed within a try()
  # function to prevent premature termination of our program when this failure occurs. The current replicate is
  # repeated until the sem() procedure successfully executes.

  ds.rm <- raw.moments(ds)   # Raw moments matrix, needed to calculate intercepts in sem() procedure

  remove("ds.sem")           # You *must* remove this object.
  remove("ds.sem.summary")   # You *must* remove this object.


  try(ds.sem <- sem(ds.model, ds.rm, n, fixed.x=c("u"), raw=TRUE),silent=TRUE)  # Estimate Model w/ SEM

  if(!exists("ds.sem")) { next }                                                #
  ds.sem.se <- sqrt(diag(ds.sem$cov))                                           # Standard Errors

  ds.sem.b <- rbind( cbind(ds.sem$coef[14], ds.sem.se[14]),
                     cbind(ds.sem$coef[15], ds.sem.se[15]))

  rownames(ds.sem.b) = rownames(b)
  try(ds.sem.summary <- summary(ds.sem), silent=TRUE)

  # Step 8: Write Results to Files
  # -------------------------------------------------------
  # Column 1: Replication Number (r = 1 to 1000)
  # Column 2: Coefficient Number 1=beta1, 2=beta2, 3=beta3, 4=gamma, 5=gamma2, 6=gamma3
  # Column 3: Coefficient Estimate
  # Column 4: Standard Error

  if( exists("ds.sem.summary") )
  {
     write(t(cbind(rep(r,2), 1:2, ds.sem.b)), ncolumn=4, file="c:\\SIMPLE_1STAGE.TXT", append=TRUE, sep="\t")
     write(t(cbind(rep(r,2), 1:2, ds.lm.b )), ncolumn=4, file="c:\\SIMPLE_2STAGE.TXT", append=TRUE, sep="\t")
     r = r + 1
  }
  else next
}

# PART TWO: THE COMPROMISE MODEL, Y = b1*t1 + b2*t2 +b3*t3 + + g1*t1*G + g2*t2*G + g3*t3*G + e
#
# The program code here is almost exactly like in the “simple model”. As such, this section of the code is not
# heavily commented like the previous section, because we are not doing anything different. The only difference
# is there are more treatment effects, and the model is slighltly different.

library(sem)
library(MASS)

ds.model <- specify.model()
G   ->  x01, lamda01, NA
G   ->  x02, lamda02, NA
G   ->  x03, lamda03, NA
G   ->  x04, lamda04, NA
G   ->  x05, lamda05, NA
G   ->  x06, lamda06, NA
G   ->  x07, lamda07, NA
G   ->  x08, lamda08, NA
G   ->  x09, lamda09, NA
```

```
G   ->  x10, lamda10, NA
G   ->  x11, lamda11, NA
G   ->  x12, lamda12, NA
G   ->  x13, lamda13, NA
y   <-    u, intercept, NA
y   <-    G, lamda_G, NA
G   <->   G, NA, 1
y   <->   y, e, NA
x01 <-> x01, delta01, NA
x02 <-> x02, delta02, NA
x03 <-> x03, delta03, NA
x04 <-> x04, delta04, NA
x05 <-> x05, delta05, NA
x06 <-> x06, delta06, NA
x07 <-> x07, delta07, NA
x08 <-> x08, delta08, NA
x09 <-> x09, delta09, NA
x10 <-> x10, delta10, NA
x11 <-> x11, delta11, NA
x12 <-> x12, delta12, NA
x13 <-> x13, delta13, NA

r <- 1
options(warn=-1)
try(temp<-read.table("c:\\FULL_1STAGE.TXT", header = FALSE, sep="\t"))
try(r <- max(temp[,1])+1)

while (r <= 10)
{

  # Step 3: Specify Regression Effects
  # ----------------------------------

  n <-90
  u <- rep(1,n)
  b <- rbind(1.5, 2.5, 3.5, 3.0, 2.0, 1.0)
  t <- cbind(c(rep(1,n/3), rep(2,n/3), rep(3,n/3)))
  rownames(b) = c("t1","t2","t3","GRIT x t1","GRIT x t2","GRIT x t3")


  # Step 4: Factor Loadings, Specific Errors, Random Errors
  # -------------------------------------------------------

  G <-rnorm(n)
  L <-rbind(0.501, 0.599, 0.591, 0.756, -0.531, 0.527, -0.643, -0.466, 0.686, 0.477, -0.531, 0.740, 0.622)
  psi   <-c(0.749, 0.641, 0.650, 0.429,  0.718, 0.723,  0.587,  0.783, 0.529, 0.772,  0.718, 0.453, 0.613)
  d <- mvrnorm(n,rep(0,13),diag(psi))
  e <- rnorm(n,0,1)

  # Step 5: Calculate X, Y. Build Final Dataset
  # -------------------------------------------

  X <- matrix(rep(0,13*n),n,13)
  Y <- rep(0,n)
  for(i in 1:n)
  {
    X[i,] <- L %*% G[i] + d[i,]
    Y[i]  <- b[1]*(t[i]==1)      + b[2]*(t[i]==2)      + b[3]*(t[i]==3) +
             b[4]*(t[i]==1)*G[i] + b[5]*(t[i]==2)*G[i] + b[6]*(t[i]==3)*G[i] + e[i]
  }
  ds <- data.frame(cbind(Y,u,X,t))
  names(ds) = cbind("y","u","x01","x02","x03","x04","x05","x06","x07","x08","x09","x10","x11","x12","x13","t")
  ds <- data.frame(cbind(ds, (ds$t==1)*1, (ds$t==2)*1, (ds$t==3)*1))
  names(ds) =  cbind("y","u","x01","x02","x03","x04","x05","x06",
                             "x07","x08","x09","x10","x11","x12","x13","t","t1","t2","t3")

  # Step 6: Two-Stage Estimation using Factor Analysis and Regression
  # -----------------------------------------------------------------

  ds.fa <- factanal(x = ds[,3:15], factors=1, scores="regression")
  ds <- data.frame(cbind(ds, ds.fa$scores))
  names(ds)[20]="g"

  ds.lm <- lm( y ~ 0 + t1 + t2 + t3 + g*t1 + g*t2 + g*t3 - g, data=ds )
  ds.lm.b <- summary(ds.lm)$coefficients[1:6,1:2]

  # Step 7: One-Stage Estimation using Structural Equation Modeling
  # ---------------------------------------------------------------
  ds.rm1 <- raw.moments(subset(ds,t==1)[,1:15]); remove("ds.sem1"); remove("ds.sem1.summary")
  ds.rm2 <- raw.moments(subset(ds,t==2)[,1:15]); remove("ds.sem2"); remove("ds.sem2.summary")
  ds.rm3 <- raw.moments(subset(ds,t==3)[,1:15]); remove("ds.sem3"); remove("ds.sem3.summary")

  try(ds.sem1 <- sem(ds.model, ds.rm1, n/3, fixed.x=c("u"), raw=TRUE),silent=TRUE)
  try(ds.sem2 <- sem(ds.model, ds.rm2, n/3, fixed.x=c("u"), raw=TRUE),silent=TRUE)
  try(ds.sem3 <- sem(ds.model, ds.rm3, n/3, fixed.x=c("u"), raw=TRUE),silent=TRUE)

  if( !(exists("ds.sem1") && exists("ds.sem2") && exists("ds.sem3")) ) { next }

  ds.sem1.se <- sqrt(diag(ds.sem1$cov))   # Standard Errors
  ds.sem2.se <- sqrt(diag(ds.sem2$cov))   #
  ds.sem3.se <- sqrt(diag(ds.sem3$cov))   #

  ds.sem.b <- rbind( cbind(ds.sem1$coef[14], ds.sem1.se[14]), cbind(ds.sem2$coef[14], ds.sem2.se[14]),
                     cbind(ds.sem3$coef[14], ds.sem3.se[14]), cbind(ds.sem1$coef[15], ds.sem1.se[15]),
                     cbind(ds.sem2$coef[15], ds.sem2.se[15]), cbind(ds.sem3$coef[15], ds.sem3.se[15]))
  rownames(ds.sem.b) = rownames(b)

  try(ds.sem1.summary <- summary(ds.sem1), silent=TRUE)
  try(ds.sem2.summary <- summary(ds.sem2), silent=TRUE)
  try(ds.sem3.summary <- summary(ds.sem3), silent=TRUE)

  # Step 8: Write Results to Files
  # -------------------------------------------------------
  # Column 1: Replication Number (r = 1 to 1000)
  # Column 2: Coefficient Number 1=beta1, 2=beta2, 3=beta3, 4=gamma, 5=gamma2, 6=gamma3
  # Column 3: Coefficient Estimate
  # Column 4: Standard Error

  if( exists("ds.sem1.summary") && exists("ds.sem2.summary") && exists("ds.sem3.summary"))
  {
     write(t(cbind(rep(r,6), 1:6, ds.sem.b)), ncolumn=4, file="c:\\FULL_1STAGE.TXT", append=TRUE, sep="\t")
     write(t(cbind(rep(r,6), 1:6, ds.lm.b )), ncolumn=4, file="c:\\FULL_2STAGE.TXT", append=TRUE, sep="\t")
     r = r + 1
  }
  else next
}

# ANALYSIS OF ASSISTMENTS DATA USING COMRPOMISE MODEL
# ---------------------------------------------------

ds.sums<-na.omit(cbind(ds.data[,c(03:04,05:16,30:38)],1))
ds.grit<-na.omit(cbind(ds.data[,c(03:04,17:29,30:38)],1))

colnames(ds.sums)[ncol(ds.sums)]="UNIT"
colnames(ds.grit)[ncol(ds.grit)]="UNIT"

# MODEL 1: The SUMS Model (Multivariate SEM)
# ------------------------------------------

ds.sums.model <- specify.model()
SUMS01 <- SUMS, lams01, NA
SUMS03 <- SUMS, lams03, NA
SUMS04 <- SUMS, lams04, NA
SUMS05 <- SUMS, lams05, NA
SUMS07 <- SUMS, lams07, NA
SUMS10 <- SUMS, lams10, NA
SUMS12 <- SUMS, lams12, NA
SUMS13 <- SUMS, lams13, NA
SUMS15 <- SUMS, lams15, NA
SUMS16 <- SUMS, lams16, NA
SUMS18 <- SUMS, lams18, NA
SUMS19 <- SUMS, lams19, NA
SUMS01 <-> SUMS01, psis01, NA
SUMS03 <-> SUMS03, psis03, NA
SUMS04 <-> SUMS04, psis04, NA
SUMS05 <-> SUMS05, psis05, NA
SUMS07 <-> SUMS07, psis07, NA
SUMS10 <-> SUMS10, psis10, NA
SUMS12 <-> SUMS12, psis12, NA
SUMS13 <-> SUMS13, psis13, NA
SUMS15 <-> SUMS15, psis15, NA
SUMS16 <-> SUMS16, psis16, NA
SUMS18 <-> SUMS18, psis18, NA
SUMS19 <-> SUMS19, psis19, NA
SUMS <-> SUMS, NA, 1
 INQ_GAIN <->  INQ_GAIN, VAR1, NA
RAMP_GAIN <-> RAMP_GAIN, VAR2, NA
```

```
 INQ_GAIN <- SUMS, GAMMA1, NA
 INQ_GAIN <- UNIT, BETA1, NA
RAMP_GAIN <- SUMS, GAMMA2, NA
RAMP_GAIN <- UNIT, BETA2, NA

ds.sums.rm1  <- raw.moments(subset(ds.sums, TREATMENT==1)[,-1])
ds.sums.rm2  <- raw.moments(subset(ds.sums, TREATMENT==2)[,-1])
ds.sums.rm3  <- raw.moments(subset(ds.sums, TREATMENT==3)[,-1])

ds.sums.n1 <- nrow(subset(ds.sums, TREATMENT==1))
ds.sums.n2 <- nrow(subset(ds.sums, TREATMENT==2))
ds.sums.n3 <- nrow(subset(ds.sums, TREATMENT==3))

ds.sums.sem1 <- sem(ds.sums.model, ds.sums.rm1, N=ds.sums.n1, fixed.x=c("UNIT"), raw=TRUE)
ds.sums.sem2 <- sem(ds.sums.model, ds.sums.rm2, N=ds.sums.n2, fixed.x=c("UNIT"), raw=TRUE)
ds.sums.sem3 <- sem(ds.sums.model, ds.sums.rm3, N=ds.sums.n3, fixed.x=c("UNIT"), raw=TRUE)

summary(ds.sums.sem1)
summary(ds.sums.sem2)
summary(ds.sums.sem3)

# MODEL 2: The GRIT Model (Multivariate SEM)
# ------------------------------------------

ds.grit.model <- specify.model()
GRIT02 <- GRIT, lamg02, NA
GRIT03 <- GRIT, lamg03, NA
GRIT04 <- GRIT, lamg04, NA
GRIT05 <- GRIT, lamg05, NA
GRIT06 <- GRIT, lamg06, NA
GRIT07 <- GRIT, lamg07, NA
GRIT08 <- GRIT, lamg08, NA
GRIT09 <- GRIT, lamg09, NA
GRIT10 <- GRIT, lamg10, NA
GRIT11 <- GRIT, lamg11, NA
GRIT12 <- GRIT, lamg12, NA
GRIT13 <- GRIT, lamg13, NA
GRIT14 <- GRIT, lamg14, NA
GRIT02 <-> GRIT02, psig02, NA
GRIT03 <-> GRIT03, psig03, NA
GRIT04 <-> GRIT04, psig04, NA
GRIT05 <-> GRIT05, psig05, NA
GRIT06 <-> GRIT06, psig06, NA
GRIT07 <-> GRIT07, psig07, NA
GRIT08 <-> GRIT08, psig08, NA
GRIT09 <-> GRIT09, psig09, NA
GRIT10 <-> GRIT10, psig10, NA
GRIT11 <-> GRIT11, psig11, NA
GRIT12 <-> GRIT12, psig12, NA
GRIT13 <-> GRIT13, psig13, NA
GRIT14 <-> GRIT14, psig14, NA
GRIT <-> GRIT, NA, 1
 INQ_GAIN <->  INQ_GAIN, VAR1, NA
RAMP_GAIN <-> RAMP_GAIN, VAR2, NA
 INQ_GAIN <- GRIT, GAMMA1, NA
 INQ_GAIN <- UNIT, BETA1, NA
RAMP_GAIN <- GRIT, GAMMA2, NA
RAMP_GAIN <- UNIT, BETA2, NA

ds.grit.rm1  <- raw.moments(subset(ds.grit, TREATMENT==1)[,-1])
ds.grit.rm2  <- raw.moments(subset(ds.grit, TREATMENT==2)[,-1])
ds.grit.rm3  <- raw.moments(subset(ds.grit, TREATMENT==3)[,-1])

ds.grit.n1 <- nrow(subset(ds.grit, TREATMENT==1))
ds.grit.n2 <- nrow(subset(ds.grit, TREATMENT==2))
ds.grit.n3 <- nrow(subset(ds.grit, TREATMENT==3))

ds.grit.sem1 <- sem(ds.grit.model, ds.grit.rm1, N=ds.grit.n1, fixed.x=c("UNIT"), raw=TRUE)
ds.grit.sem2 <- sem(ds.grit.model, ds.grit.rm2, N=ds.grit.n2, fixed.x=c("UNIT"), raw=TRUE)
ds.grit.sem3 <- sem(ds.grit.model, ds.grit.rm3, N=ds.grit.n3, fixed.x=c("UNIT"), raw=TRUE)

summary(ds.grit.sem1)
summary(ds.grit.sem2)
summary(ds.grit.sem3)
```

APPENDIX B: SURVEY QUESTIONS

Item	Survey Text
SUMS01	A model is a smaller sized version of an object.
SUMS03	Models are used to make predictions about a scientific object or process.
SUMS04	Models show the relationship between ideas.
SUMS05	Models are used to help test theories about a scientific object or process.
SUMS06	A model may be changed if there are new ideas about a scientific object or process.
SUMS07	Models are used to help formulate ideas about a scientific objector process.
SUMS08	More than one model is used to show different versions of an object or process.
SUMS09	A model may be changed if there are new findings about a scientific object or process.
SUMS10	Models are used to explain a scientific object or process.
SUMS11	A model may be changed if there are new theories about a scientific object or process.
SUMS12	A model is similar to the real thing in every way except for size.
SUMS13	Models are used to show an object or a process in a visual way.
SUMS14	More than one model is used to show different perspectives about an object or process.
SUMS15	A model shows what the real thing looks like.
SUMS16	A model includes what is needed to show or explain a scientific object or process.
SUMS17	More than one model is used to show different sides or features of an object or process.
SUMS18	Models are used to help test predictions about a scientific object or process.
SUMS19	A model is an exact replica of the real thing.
SUMS20	There may be more than one model for an object or process because different people may have different viewpoints about what it looks like or how it works.
SUMS21	A model may be changed if new technology allows us to discover new evidence about an object or process.
SUMS22	More than one model may be used if different people interpret the data/evidence in different ways.

Item	Survey Text
GRIT01	I have a hard time breaking bad habits.
GRIT02	I am lazy.
GRIT03	I say inappropriate things.
GRIT04	I do certain things that are bad for me, if they are fun.
GRIT05	I refuse things that are bad for me.
GRIT06	I wish I had more self-discipline.
GRIT07	I am good at resisting temptation.
GRIT08	People would say that I have iron self-discipline.
GRIT09	Pleasure and fun sometimes keep me from getting work done.
GRIT10	I have trouble concentrating.
GRIT11	I am able to work effectively towards long-term goals.
GRIT12	Sometimes I can't stop myself from doing something, even if I know it is wrong.
GRIT13	I often act without thinking through all the alternatives.