



All Theses and Dissertations

2011-06-27

Hierarchical Probit Models for Ordinal Ratings Data

Allison M. Butler

Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Statistics and Probability Commons](#)

BYU ScholarsArchive Citation

Butler, Allison M., "Hierarchical Probit Models for Ordinal Ratings Data" (2011). *All Theses and Dissertations*. 2656.
<https://scholarsarchive.byu.edu/etd/2656>

This Selected Project is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

Hierarchical Probit Models for Ordinal Ratings Data

Allison M. Butler

A project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Natalie J. Blades, Chair
Bruce J. Collings
John S. Lawson

Department of Statistics
Brigham Young University

August 2011

Copyright © 2011 Allison M. Butler

All Rights Reserved

ABSTRACT

Hierarchical Probit Models for Ordinal Ratings Data

Allison M. Butler
Department of Statistics, BYU
Master of Science

University students often complete evaluations of their courses and instructors. The evaluation tool typically contains questions about the course and the instructor on an ordinal Likert scale. We assess instructor effectiveness while adjusting for known confounders. We present a probit regression model with a latent variable to measure the instructor effectiveness accounting for student specific covariates, such as student grade in the course, high school and university GPA, and ACT score.

Keywords: Bayesian, probit, ordinal

ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my advisor, Dr. Blades, for her unceasing support on my masters project. I am grateful for her guidance, insights, and advice. I could not have completed this project without her knowledge, counsel, and support.

I would also like to thank my parents for their unfailing support. I am deeply grateful for their advice, encouragement, and love. I would not be who and where I am today without them.

CONTENTS

Contents	iv
1 Introduction	1
1.1 Review of Methods	1
2 Hierarchical Probit Model of Instructor Effectiveness	5
2.1 Likert Scales and Latent Variables	5
2.2 Bayesian Probit Model with Latent Variable	7
2.3 Hierarchical Probit Model with Latent Variable	8
3 Dichotomous Data Analysis	11
3.1 Data	12
3.2 Prior Selection	12
3.3 Posterior Distributions	13
4 Ordinal Hierarchical Probit Model of Instructor Effectiveness	23
4.1 Cutpoint Structure	23
4.2 Ordinal Hierarchical Probit Model with Latent Variable	24
5 Polychotomous Data Analysis	28
5.1 Data	28
5.2 Prior Selection	28
5.3 Posterior Distributions	29
6 Conclusions	41

Bibliography	43
Appendices	44
Appendix A: Derivation of Complete Conditional Distributions	45
A.1 Hierarchical Probit Model	45
A.2 Ordinal Hierarchical Probit Model	48
Appendix B: Mixing Plots	50
B.1 Mixing Plots for Binary Hierarchical Probit Model	50
B.2 Mixing Plots for Ordinal Hierarchical Probit Model	55
Appendix C: R Code	60
C.1 Code for Hierarchical Probit Model	60
C.2 Code for Ordinal Hierarchical Probit Model	78

INTRODUCTION

People are often asked to rate their experiences with a new product, a presentation, a movie, music, etc. These ratings are usually collected on an ordinal scale, such as 1 to 5, or with quality words, such as from strongly disagree to strongly agree. For example, Netflix, the popular movie rental website, asks their customers to rate movies from one star to five stars in an effort to suggest movies specific to each customers preferences.

Another common source of ordinal ratings data is student evaluations. University students are often asked to complete evaluations for their instructors. These evaluations are used to assess instructor effectiveness; however, the student evaluations indicate the student-perceived instructor effectiveness, not the true instructor effectiveness. Marsh (1987) presents a literature review of studies reporting correlation between student ratings and prior subject interest, expected grades, and workload or course difficulty. In order to assess true instructor effectiveness we must adjust for known confounders such as these.

Ordinal ratings data are very common and come with their own set of statistical challenges. While ordinal ratings data are pervasive, the non-numeric nature of these data obviates the use of many simple analyses. We will review methods for analyzing such ordinal data by first considering the simplest case—the two-level ordinal dichotomy. We then discuss methods for extending these models from dichotomous orderings to polytomous orderings.

1.1 REVIEW OF METHODS

Logistic Regression

The relationship between a binary response and one or more predictors is often described with logistic regression, which models the probability that the response equals one given the

predictors. We model the log odds ratio:

$$\ln \left(\frac{P(y = 1 | \mathbf{X} = \mathbf{x})}{P(y = 0 | \mathbf{X} = \mathbf{x})} \right) = \mathbf{x}'\boldsymbol{\beta}. \quad (1.1)$$

Transformation of the model yields the probability that $y = 1$:

$$P(y = 1 | \mathbf{X} = \mathbf{x}) = \frac{e^{\mathbf{x}'\boldsymbol{\beta}}}{1 + e^{\mathbf{x}'\boldsymbol{\beta}}}. \quad (1.2)$$

Standard software makes this model easily accessible. Based on the canned output, a rule could easily be created to classify each observation as a poor instructor or a good instructor by choosing an appropriate cutoff probability.

While this model shares some simple structure with the general linear model, the coefficients are not as easily interpreted. The estimated β 's must be interpreted in terms of log odds or log odds ratios. The logistic regression model explains how the confounders affect the student evaluations. However, it does not estimate instructor effectiveness after adjusting for confounders.

Bayesian Logistic Model

In frequentist statistics we assume our parameters are fixed, but unknown. This means that we only get point estimates of our parameters, and confidence intervals are based on the accuracy of the method and not on the parameter. In the Bayesian paradigm we assume the parameters are unknown but variable. This means that we get entire distributions of parameter estimates and we can measure the variability of the estimates. Credible intervals are now intuitive and can be interpreted as the probability that the parameter lies in that interval range.

To place the logistic regression model in a Bayesian framework, we start with the same likelihood function for the data: with a binary outcome, we use the binomial distribution,

$$f(y_i | p_i) = \binom{N}{y_i} p_i^{y_i} (1 - p_i)^{N - y_i}, \quad (1.3)$$

where i is the student who completed the evaluation, y_i is the number of favorable ratings given by student i , p_i is the probability of student i giving a favorable rating, and N is the number of ratings given by student i . In our case, $N = 1$ because each student only rates the instructor once, so the likelihood function is,

$$L(p_i|y_i) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}, \quad (1.4)$$

where n is the number of students completing an evaluation. To link the probability of success, p_i , to the predictor variables, we use a link function so that,

$$p_i = F(\mathbf{x}'_i \boldsymbol{\beta}). \quad (1.5)$$

As in the logistic regression framework above, the logistic or logit link function yields,

$$p_i = \frac{e^{\mathbf{x}'_i \boldsymbol{\beta}}}{1 + e^{\mathbf{x}'_i \boldsymbol{\beta}}}. \quad (1.6)$$

Using this link function, we can rewrite the likelihood function in 1.4 as

$$L(\boldsymbol{\beta}|y_i) = \prod_{i=1}^n \left(\frac{e^{\mathbf{x}'_i \boldsymbol{\beta}}}{1 + e^{\mathbf{x}'_i \boldsymbol{\beta}}} \right)^{y_i} \left(1 - \frac{e^{\mathbf{x}'_i \boldsymbol{\beta}}}{1 + e^{\mathbf{x}'_i \boldsymbol{\beta}}} \right)^{1-y_i}. \quad (1.7)$$

A typical choice of prior for $\boldsymbol{\beta}$ specifies that $\boldsymbol{\beta}$ comes from a multivariate normal distribution with mean vector \mathbf{b} and covariance matrix \mathbf{B} ,

$$\boldsymbol{\beta} \sim N(\mathbf{b}, \mathbf{B}). \quad (1.8)$$

This choice of likelihood and prior yields the joint posterior distribution,

$$\pi(\boldsymbol{\beta}|y) \propto \pi(\boldsymbol{\beta}) \times \prod_{i=1}^n \left(\frac{e^{\mathbf{x}\boldsymbol{\beta}}}{1 + e^{\mathbf{x}\boldsymbol{\beta}}} \right)^{y_i} \left(1 - \frac{e^{\mathbf{x}\boldsymbol{\beta}}}{1 + e^{\mathbf{x}\boldsymbol{\beta}}} \right)^{1-y_i}; \quad (1.9)$$

however, this will be computationally complicated since the complete conditional distributions will not have closed forms (Johnson and Albert 1999).

The complete conditional distributions are the distribution of one parameter in the model given all other parameters in the model and the data, \mathbf{y} . For example, suppose the model contains parameters $\boldsymbol{\theta} = (\theta_1, \dots, \theta_k)'$. The complete conditional distributions are $\pi(\theta_i | \boldsymbol{\theta}_{i \neq j}, \mathbf{y})$ (Carlin and Louis 2009).

Bayesian Probit Model

Another approach would be to use the probit link function instead of the logit link function.

That is,

$$p_i = \Phi(\mathbf{x}'_i\boldsymbol{\beta}), \quad (1.10)$$

where Φ is the distribution function of the standard normal distribution. The probit link function yields the likelihood function,

$$L(\boldsymbol{\beta}|y_i) = \prod_{i=1}^n \left(\Phi(\mathbf{x}'_i\boldsymbol{\beta}) \right)^{y_i} \left(1 - \Phi(\mathbf{x}'_i\boldsymbol{\beta}) \right)^{1-y_i}. \quad (1.11)$$

Again, by choosing a multivariate normal distribution for $\boldsymbol{\beta}$ the joint posterior distribution is

$$\pi(\boldsymbol{\beta}|y_i) \propto \pi(\boldsymbol{\beta}) \times \prod_{i=1}^n \left(\Phi(\mathbf{x}'_i\boldsymbol{\beta}) \right)^{y_i} \left(1 - \Phi(\mathbf{x}'_i\boldsymbol{\beta}) \right)^{1-y_i}. \quad (1.12)$$

Unfortunately, this model is still computationally complicated as the complete conditional distributions do not have closed forms. As in logistic regression, this model allows us to explain how the confounders affect the student evaluations, except we have posterior distributions on $\boldsymbol{\beta}$ and not just point estimates. However, we really want to make conclusions about the true instructor effectiveness. We do not have any direct measurements on true instructor effectiveness so we will need to introduce a latent variable. Chapter 2 will discuss one possible model that uses a Bayesian framework to incorporate a latent variable.

HIERARCHICAL PROBIT MODEL OF INSTRUCTOR EFFECTIVENESS

Albert and Chib (1993) introduce a data augmentation scheme which makes the Bayesian probit model much easier to fit. They create a system in which it is easy to simulate draws from the posterior distribution via Gibbs sampling. In this chapter we discuss this model in greater detail in the context of student evaluation data. We will discuss an extension of this model for ordinal data in Chapter 4.

We receive student evaluated instructor effectiveness data on an ordinal scale from 1 to 8; however the use of the Likert scale is for the convenience of the evaluators. Thus, it seems reasonable to imagine true instructor effectiveness is a continuous latent variable. In this chapter, we discuss Likert scales and latent variables, a common binary probit model, and a hierarchical probit model.

2.1 LIKERT SCALES AND LATENT VARIABLES

Figure 2.1 gives an example of a latent variable distribution for instructor effectiveness. The underlying instructor effectiveness is continuously distributed—poor instructors fall somewhere on the left hand side of the distribution, while good instructors fall somewhere on the right hand side of the distribution. An 8-level Likert scale creates cutoffs in the distribution, so the observations we collect tell us what range the student perception of the instructor falls within. For example, if an instructor falls between the upper and lower cutoff for 7, we will observe a student rating of 7.

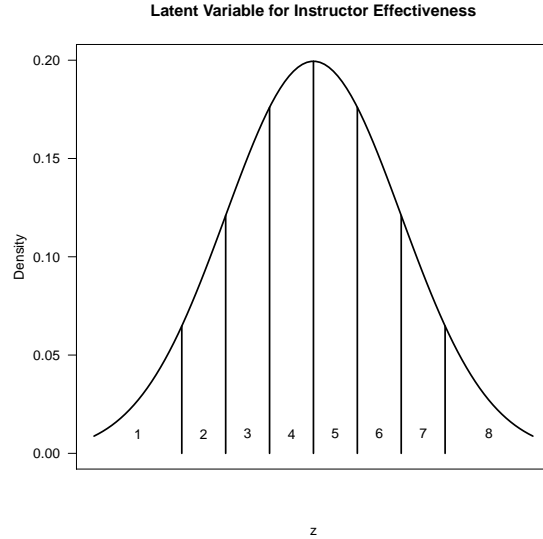


Figure 2.1: Example of latent structure of true instructor effectiveness. True instructor effectiveness is continuously distributed, but we only observe the intervals created by the cutoffs of that distribution as student ratings from 1 to 8.

Each student evaluates instructor effectiveness differently, so each student has different cutoffs which determine the rating the instructor will receive. Figure 2.2 shows two possible examples of student evaluated instructor effectiveness. Figure 2.2(a) depicts the cutoffs for a student who tends to evaluate high, because there is a wide range of effectiveness that would be categorized as a 6, 7, or 8. In fact, even instructors just below the median of instructor effectiveness would get at least a 6. On the other hand, Figure 2.2(b) depicts the cutoffs for a student who tends to evaluate low. An instructor must be truly exceptional to receive a 6, 7, or 8 because the range of effectiveness that would receive a 6, 7, or 8 is very small.

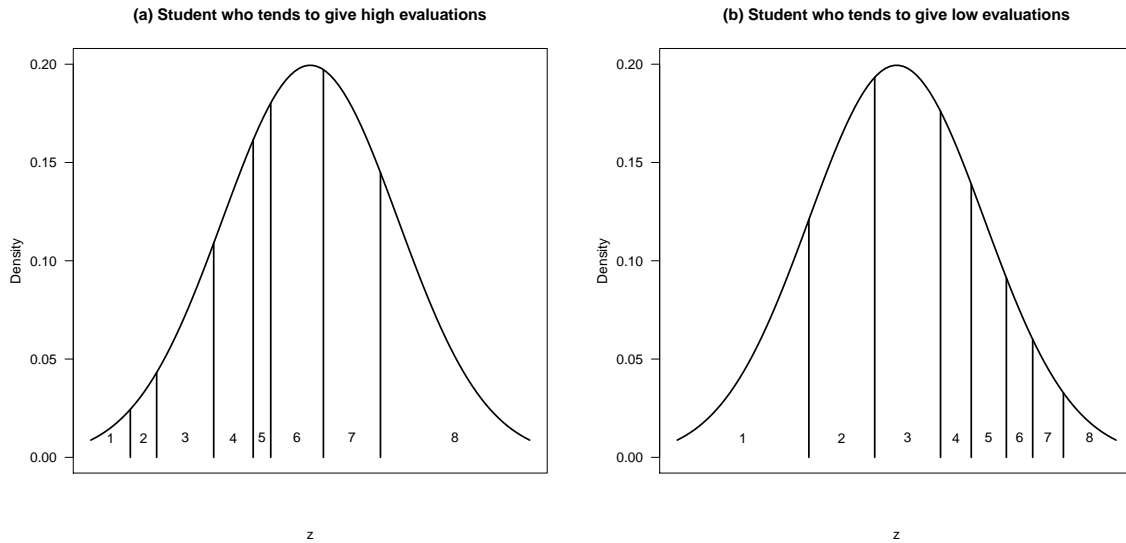


Figure 2.2: Example of latent structure of true instructor effectiveness for two different students. Figure (a) depicts a student who tends to give high ratings because there is a large range for 6, 7, and 8. Figure (b) depicts a student who tends to give low ratings because there is a small range for 6, 7, and 8.

2.2 BAYESIAN PROBIT MODEL WITH LATENT VARIABLE

Before tackling the task of estimating all the cutoffs for an 8-level Likert scale, we consider simplifying the response to a binary variable with only one cutoff value. To create a binary response from the 8-level Likert data, we set the response to zero if the rating is from 1 to 5, and set the response to one if the rating is from 6 to 8. Figure 2.3 depicts the effect of simplifying the response to a binary variable on the latent variable distribution. An observation below the cutoff receives a 0 and an observation above the cutoff receives a 1.

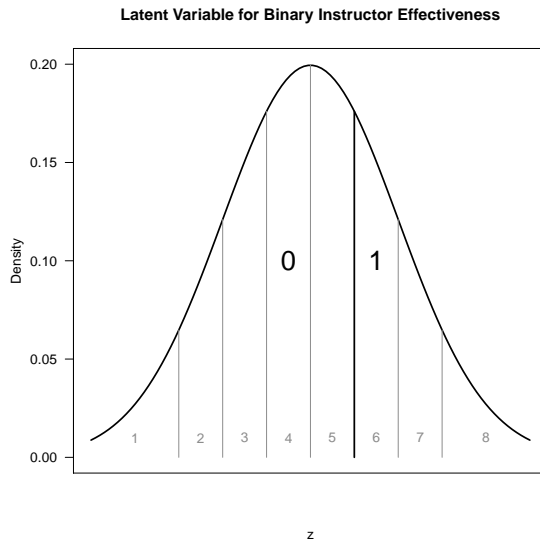


Figure 2.3: Example of latent structure of true instructor effectiveness for binary response.

Following [Albert and Chib \(1993\)](#) we introduce z_1, \dots, z_n where $z_i \sim N(\mathbf{x}'_i\boldsymbol{\beta}, 1)$ and $y_i = 1$ when $z_i > 0$ and $y_i = 0$ when $z_i \leq 0$. We link the latent instructor effectiveness to the predictors linearly, that is $z_i = \mathbf{x}'_i\boldsymbol{\beta} + \epsilon_i$, where $\epsilon_i \sim N(0, 1)$.

As in the previous Bayesian models, $\boldsymbol{\beta}$ has a multivariate normal prior, so the joint posterior distribution is

$$\pi(\boldsymbol{\beta}, \mathbf{z}|\mathbf{y}) \propto \pi(\boldsymbol{\beta}) \prod_{i=1}^n \varphi(z_i - \mathbf{x}'_i\boldsymbol{\beta}) [I(z_i > 0)^{y_i} + I(z_i \leq 0)^{1-y_i}], \quad (2.1)$$

where $\varphi(\cdot)$ is the standard normal density and $I(\cdot)$ is an indicator function.

This model is computationally tractable as the complete conditional distributions have closed form solutions ([Johnson and Albert 1999](#)), and we have a latent variable that helps us to understand true instructor effectiveness. Yet, we are not incorporating the inherent hierarchical structure of the data by allowing for instructor and course differences.

2.3 HIERARCHICAL PROBIT MODEL WITH LATENT VARIABLE

In order to account for the differences in instructor and course we will modify the latent variable structure by including an instructor by course term as well as the student confounders.

The latent variable structure is

$$z_{ijk} = \gamma_{ij} + x'_{ijk}\boldsymbol{\beta} + \epsilon_{ijk}, \quad (2.2)$$

where z_{ijk} is student k 's latent instructor effectiveness assessment for instructor i teaching course j , γ_{ij} is the effect of instructor i teaching course j , $\boldsymbol{\beta}$ holds the coefficients for the student predictors, x_{ijk} is a vector of student k 's predictors from course j taught by instructor i , and the errors, ϵ_{ijk} , are distributed $N(0, 1)$.

The likelihood function is still binomial; however, we change the indices to represent the instructor, course, student relationship. The likelihood function is,

$$L(p_{ijk}|y_{ijk}) = \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} [(p_{ijk})^{y_{ijk}} (1 - p_{ijk})^{1-y_{ijk}}], \quad (2.3)$$

where n is the number of instructors, n_i is the number of courses taught by instructor i , and n_{ij} is the number of students who completed evaluations for instructor i teaching course j .

We will still use the probit link function to connect the probability of a student giving a favorable rating to the latent variable and confounders,

$$p_{ijk} = \Phi(z_{ijk}). \quad (2.4)$$

As above, the prior distribution on $\boldsymbol{\beta}$ will be multivariate normal, $\boldsymbol{\beta} \sim N(\mathbf{b}, \mathbf{B})$, with density function

$$\pi(\boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi\mathbf{B}}} e^{-\frac{(\boldsymbol{\beta}-\mathbf{b})^2}{2\mathbf{B}}}. \quad (2.5)$$

The prior distribution for the instructor by course effect, will be normal, $\gamma_{ij} \sim N(\mu_i, \sigma_i^2)$, with density function

$$\pi(\gamma_{ij}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}}. \quad (2.6)$$

The hyperpriors for γ_{ij} are chosen so that μ_i comes from a normal distribution and σ_i^2 comes from an inverse gamma distribution. That is, $\mu_i \sim N(m, s^2)$ and $\sigma_i^2 \sim IG(a, A)$, with density functions

$$\pi(\mu_i) = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(\mu_i-m)^2}{2s^2}} \quad (2.7)$$

and

$$\pi(\sigma_i^2) = \frac{1}{\Gamma(a)A^a} (\sigma_i^2)^{-(a+1)} e^{\frac{-1}{A\sigma_i^2}}. \quad (2.8)$$

The joint posterior distribution is,

$$\pi(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \mathbf{y}) \propto \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \left[\pi(\boldsymbol{\beta}) \pi(\mu_i) \pi(\sigma_i^2) \pi(\gamma_{ij}) \times \right. \quad (2.9)$$

$$\left. \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) [I(z_{ijk} > 0)^{y_{ijk}} + I(z_{ijk} \leq 0)^{1-y_{ijk}}] \right]. \quad (2.10)$$

As the complete conditional distributions are available in closed form, a Gibbs Sampler provides draws from the joint posterior distribution. The complete conditional distributions follow:

$$[z_{ijk}] \sim N(\gamma_{ij} + x'_{ijk} \boldsymbol{\beta}, 1), \quad (2.11)$$

truncated on the right at 0 if $y_{ijk} = 0$ and truncated on the left at 0 if $y_{ijk} = 1$,

$$[\boldsymbol{\beta}] \sim N([\mathbf{B}^{-1} + \mathbf{x}'\mathbf{x}]^{-1}[\mathbf{B}^{-1}\mathbf{b} + \mathbf{x}'(\mathbf{z} - \boldsymbol{\gamma})], [\mathbf{B}^{-1} + \mathbf{x}'\mathbf{x}]^{-1}), \quad (2.12)$$

$$[\gamma_{ij}] \sim N\left(\frac{\mu_i + \sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta})}{1 + \sigma_i^2 n_{ij}}, \frac{\sigma_i^2}{1 + \sigma_i^2 n_{ij}}\right), \quad (2.13)$$

$$[\mu_i] \sim N\left(\frac{\sigma_i^2 m + s^2 \sum_{j=1}^{n_i} \gamma_{ij}}{\sigma_i^2 + s^2 n_i}, \frac{\sigma_i^2 s^2}{\sigma_i^2 + s^2 n_i}\right), \quad (2.14)$$

and

$$[\sigma_i^2] \sim IG\left(a + \frac{n_i}{2}, \left[\frac{1}{A} + \frac{\sum_{j=1}^{n_i} (\gamma_{ij} + \mu_i)^2}{2}\right]^{-1}\right). \quad (2.15)$$

The following chapter contains the application of this model to a portion of the student ratings data from BYU.

DICHOTOMOUS DATA ANALYSIS

The students at Brigham Young University are encouraged to complete student evaluations for each course each semester. The student evaluation survey consists of 21 questions about the course and the instructor on an 8-level Likert scale. The instructors use these evaluations to improve their courses. Department chairs and university committees use the evaluations to help assess instructor effectiveness and determine who is retained and promoted.

We are interested in assessing true instructor effectiveness and will restrict our attention to the evaluation of the instructors and not the courses. The student evaluations tell us the student-perceived instructor effectiveness, not the true instructor effectiveness. Simply reporting the average of the student evaluations is inadequate; in order to better assess true instructor effectiveness we must adjust for confounding factors.

The data contain student-level and course-level confounding factors. Student specific covariates are blinded student ID, student grade in the course, high school and university GPA, student major, age, marital status, gender, and class standing. Course-level variables are credit hours, class size, instructor status, term, year, college, and department.

Using the hierarchical probit model introduced in Section 2.3, we can adjust for the known confounders, and we have the ability to distinguish good instructors from poor instructors. This will yield an estimate of instructor effectiveness that can be used to reward or promote the good instructors and advise or remove the poor instructors. In this chapter we use the dichotomous ordinal model to distinguish the good instructors from the poor instructors; consequently, as in Section 2.2, we will change the 8-level student ratings to a binary response. Ratings from 1 to 5 are mapped to zero, and ratings from 6 to 8 are mapped to one.

3.1 DATA

The data set we consider contains 868 student evaluations collected in a single department over five consecutive years. There are seven instructors in this department collectively teaching 22 courses.

For this analysis we include class size, student grade in the class, student total GPA, and the interaction between student grade and GPA. Student grade in the class was changed to the corresponding grade points used in the calculation of GPA so that grade could be treated as a continuous variable.

3.2 PRIOR SELECTION

While the distributional form of each prior was outlined in Section 2.3, full specification of the model requires a reasonable choice of hyperparameters for the priors on μ_i , σ_i^2 , and β . The mean of γ_{ij} , that is, μ_i , can be thought of as the true ability of each instructor to teach. We would assume that most instructors at BYU have a high ability to teach, so we center the prior distribution at one and give a fairly large variability; specifically

$$\mu_i \sim N(1, 3). \tag{3.1}$$

The variance of γ_{ij} , σ_i^2 , can be thought of as the variability in the true teaching ability of an instructor within all of the courses taught by that instructor. We would assume the variability within an instructor is quite small (depending on the courses being taught), so we choose a distribution with 97% of the mass between 0 and 1,

$$\sigma_i^2 \sim IG(10, 5). \tag{3.2}$$

The coefficients for the student predictors, β , are class size, student grade in the class, student GPA, and the interaction between grade and GPA. As class size increases, the student generally has less interaction with the instructor, and we would expect that the student rating would decrease. Thus, we select a prior for β_{size} with a mean of -0.5. With

a probit link function, a regression coefficient is interpreted as a z -score, so as class size increases by one student the z -score decreases by -0.5.

If a student gets a good grade in a course, we would assume the student learned a lot, meaning the instructor did a good job teaching the class. As student grade increases, the student ratings should increase. A prior mean of 1 says that as student grade increases by one point the z -score will increase by 1.

Students with high GPA's are typically good students who get good grades. It seems reasonable to assume these students would tend to rate their instructors high. We select a prior mean of β_{GPA} of 2, which means that as student GPA increases by one point the z -score will increase by 2.

We also suspect that there may be an interaction between grade and GPA. For example, a student who generally does not do well in classes (has a low GPA) who gets a good grade in this class may be inclined to rate the instructor high. On the other hand, a student who generally does very well in classes (has a high GPA) and gets a bad grade in this course may be inclined to rate the instructor poorly. The prior mean of β_{int} is 0.5, thus the z -score increases by 0.5 as the interaction of grade and GPA increases by one.

We will give a fairly large variability of four to each β . In summary, the prior distribution for the regression component vector is

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{size} \\ \beta_{grade} \\ \beta_{GPA} \\ \beta_{int} \end{bmatrix} \sim N \left(\begin{bmatrix} -0.5 \\ 1 \\ 2 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \right). \quad (3.3)$$

3.3 POSTERIOR DISTRIBUTIONS

The Gibbs Sampler yields posterior draws of μ_i , σ_i^2 , γ_{ij} , z_{ijk} , and $\boldsymbol{\beta}$. The draws of μ_i and σ_i^2 tell us about each instructor's teaching ability and variability in their teaching ability. The draws from γ_{ij} give us a way to compare the courses taught by the same instructor

and the instructors who taught the same course. The draws from z_{ijk} can be used to examine individual students' latent distributions for determining instructor effectiveness. The β 's describe the relationship between the confounders and the probability of receiving a favorable rating.

Posterior Distributions of μ_i and σ_i^2

The posterior distributions for μ_i represent the data-based assessment of the teaching ability of each instructor, and the posterior distributions for σ_i^2 represent the variability in assessed teaching ability of each instructor within all of the courses taught by that instructor.

Figure 3.1 shows the posterior distributions for μ_i and σ_i^2 . When looking at the posterior distributions of μ_i we notice that all the instructors seem to have very similar distributions except for instructor 3086. The teaching ability of instructor 3086 seems to be below the rest of the instructors. When looking at the posterior distributions for σ_i^2 we find that instructor 3086 seems to have a large variability in teaching ability.

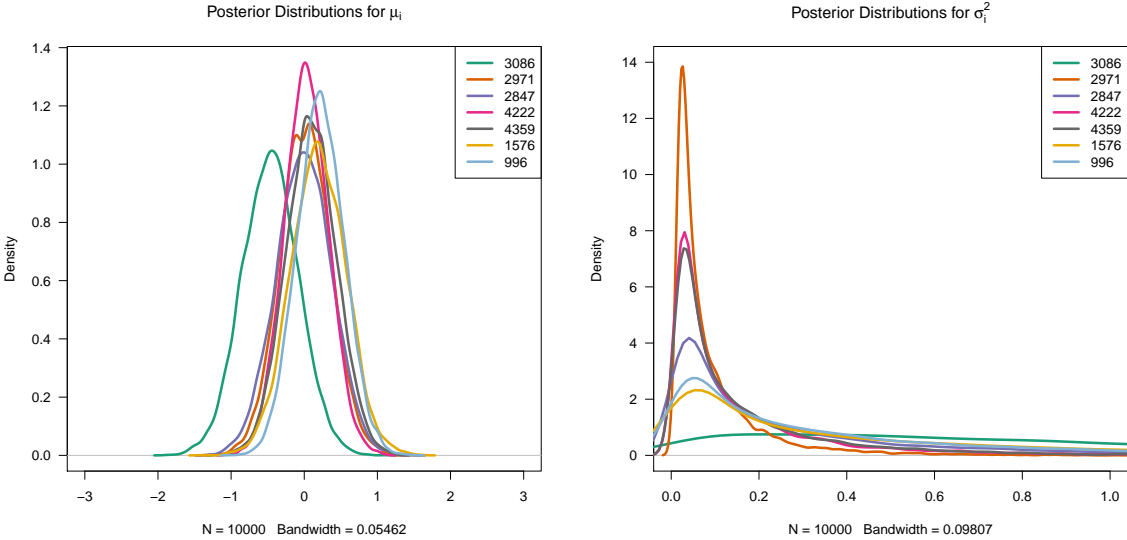


Figure 3.1: Posterior distributions for μ_i , true instructor teaching ability, and σ_i^2 , variability in instructor teaching ability.

Table 3.1 shows the unadjusted mean student ratings, the percent of student ratings that were favorable, and the standard deviation of student ratings for each instructor. Comparing this to the results from the posterior distributions of μ_i and σ_i^2 we find that instructor 3086 has very low mean score of 4.09 with only 20% of the ratings between 6 and 8. Instructor 3086 also has the highest standard deviation.

Table 3.1: Summary statistics of student ratings for each instructor.

Instructor	Overall Mean Score	Favorable Ratings	Standard Deviation
3086	4.09	20%	0.79
2971	5.83	61%	0.34
2847	5.88	62%	0.13
4222	5.88	62%	0.52
4359	6.34	77%	0.18
1576	6.36	80%	0.27
996	7.13	90%	0.25

We were able to identify an instructor that seems to be less effective, but were unable to identify the outstanding instructors. Instructor 996 seems to be an exceptional teacher with a mean score of 7.13 and 90% of the ratings between 6 and 8. However, since we have simplified the response to a binary variable we only have the power to distinguish instructors who get lots of zeros from instructors who get lots of ones. Instructor 3086 is generally in the zero range while the rest of the instructors are generally in the one range.

Posterior Distributions of γ_{ij}

Next, we look at the posterior distributions of γ_{ij} . First, we plot the posterior distributions for an instructor to compare an instructor's teaching ability in different courses. Figure 3.2 shows the lowest rated teacher, instructor 3086, and the highest rated teacher, instructor 996. We can see that there is some variability from course to course for each instructor.

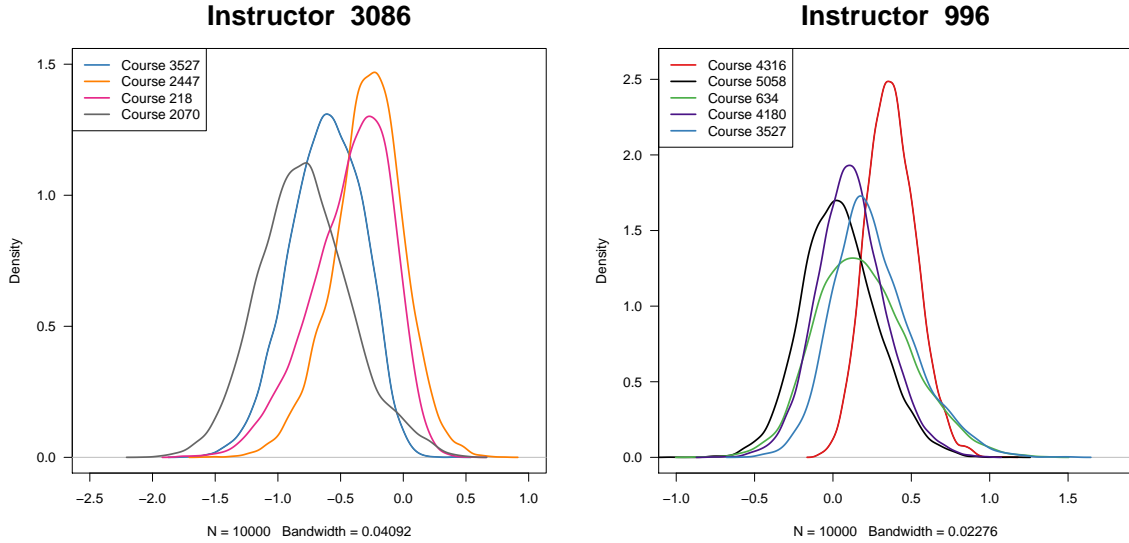


Figure 3.2: Posterior distributions for γ_{ij} by instructor, which allow us to examine the course to course variability within an instructor.

We can also plot the posterior distributions for γ_{ij} for a course and compare instructors who taught the same course. Figure 3.3 shows all the courses that were taught by multiple instructors. In most of the courses the instructors seem to have similar ability to teach. However, we can see that instructor 3086, who has the lowest mean score, has a much lower teaching ability in courses 3527 and 2070 than the other instructors. Instructor 996, who has the highest mean score, teaches courses 3527 and 634; however, we do not find differences in teaching ability with the other instructors teaching those courses. Because we have simplified the response to binary, we still cannot distinguish instructor 996 as exceptional in courses 3527 or 634.

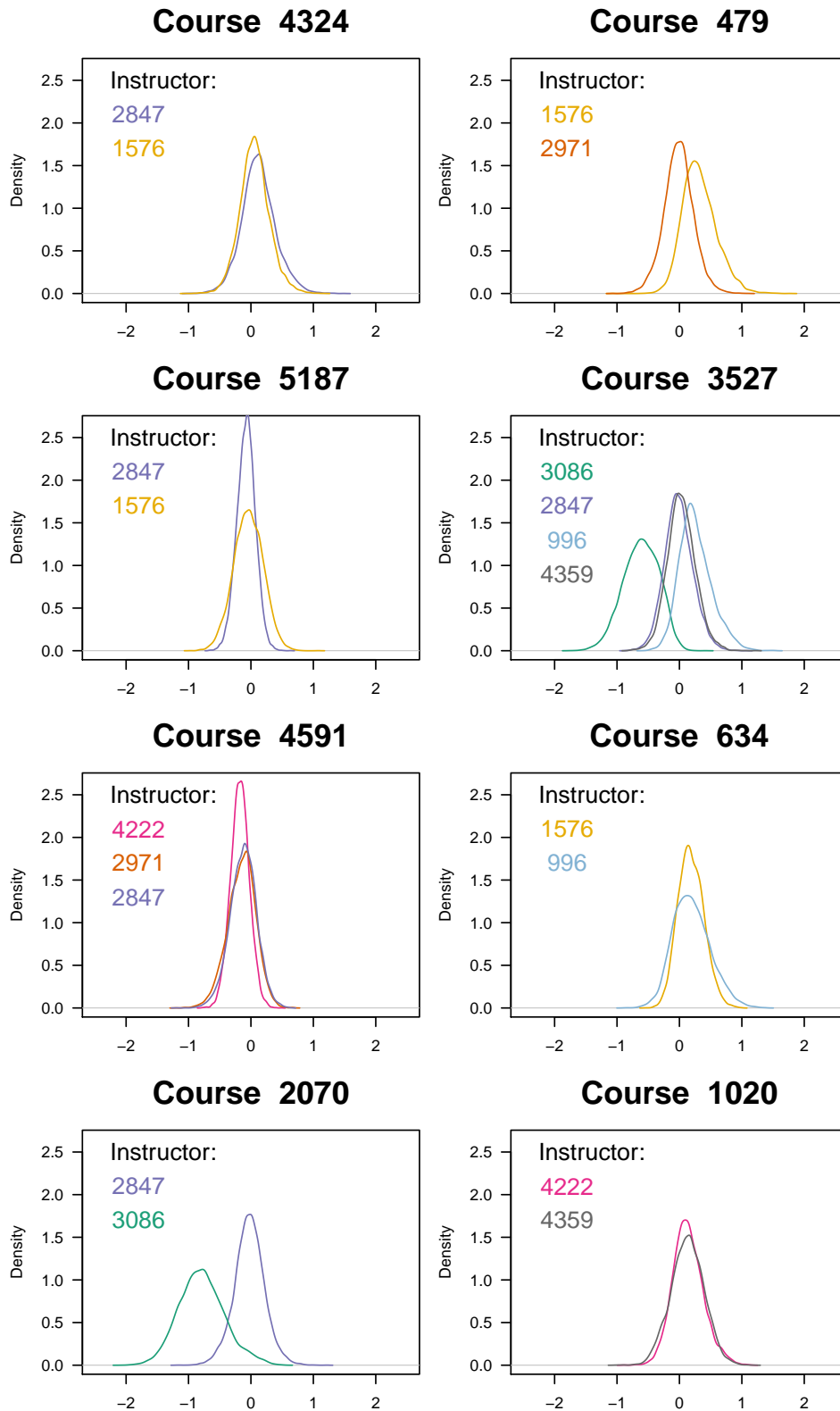


Figure 3.3: Posterior distributions of γ_{ij} for all courses taught by two or more instructors, which enables us to compare instructors who taught the same course.

Posterior Distributions of z_{ijk}

We also look at the latent variables, z_{ijk} . These latent variables measure the latent assessment of instructor i 's effectiveness in course j as perceived by student k . Recall that the cutoff value is zero, so any z_{ijk} greater than zero would rate the instructor a 1. Figure 3.4 shows the latent variable posterior distributions for each student in course 3527 by instructor. With instructor 3086, only two students have latent variable distributions that have a high probability of rating instructor 3086 favorably. However, all but one of the students who rated instructor 996 has a high probability of rating instructor 996 favorably.

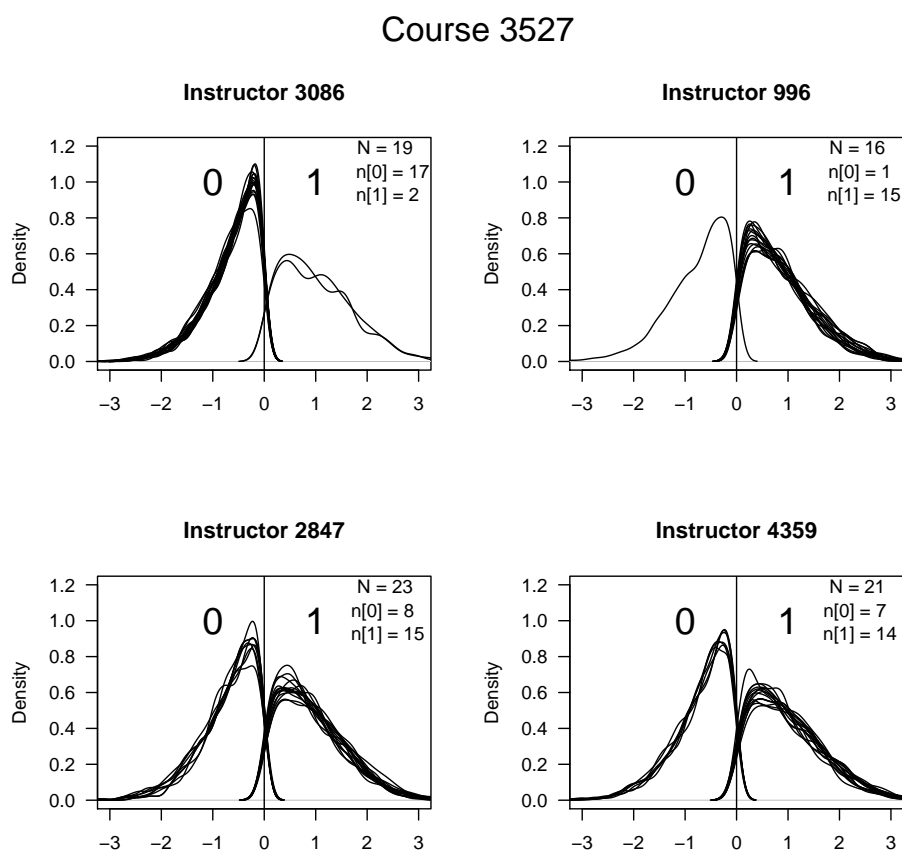


Figure 3.4: Posterior distributions for the latent variable, z_{ijk} , for each instructor who taught course 3527. For each student we are able to examine the probability of rating the instructor favorably.

Posterior Distributions for β

Lastly, we look at the student predictors, β . The posterior distributions for each β are approximately normally distributed. Table 3.2 reports the posterior means for each β as well as the 95% central credible interval (CI). If the 95% CI does not contain zero, we say that the coefficient is significant. The coefficient for class size, β_{size} , is not significant, thus class size does not seem to affect how a student rates an instructor. However, these data contain a small range of class sizes, only 10–55 students, and we may see a greater influence in class size when we have class sizes in the hundreds. The coefficients for grade, GPA, and the interaction between grade and GPA are significant, so the combination of grade and GPA does affect how a student rates an instructor.

As mentioned in Section 5.2, probit model regression coefficients, β , can be interpreted as z -scores. As class size increases by one student the z -score decreases by -0.0077. As student grade increases by one point the z -score increases by 0.4189. As student GPA increases by one point the z -score increases by 0.2589. As the interaction between student grade and GPA increases by one point the z -score decreases by 0.1388.

Table 3.2: The posterior mean for each β with the 95% central credible interval.

	$E(\cdot \mathbf{y})$	2.5%	97.5%
β_{size}	-0.0077	-0.0166	0.0015
β_{grade}	0.4189	0.1698	0.6757
β_{GPA}	0.2589	0.1002	0.4182
β_{int}	-0.1388	-0.2145	-0.0630

Predictive Distributions

To ensure our prior selections result in a reasonable sample of student evaluations, we examine the prior predictive distribution. We generate 10,000 draws from β , μ_i , and σ_i^2 . Draws

from μ_i and σ_i^2 are used to generate 10,000 draws from γ_{ij} . Then, p_{ijk} is calculated as,

$$p_{ijk} = \varphi(\gamma_{ij} + \mathbf{x}'_{ijk}\boldsymbol{\beta}). \quad (3.4)$$

We can then use those p_{ijk} to generate 10,000 draws from the likelihood and see what a sample from that distribution would look like.

Figure 3.5 shows the prior predictive distribution for different levels of the confounders. We have small, medium, and large class sizes, grades A through E, and GPA from 1.0 to 4.0. The vertical axis shows the proportion of students who rated the instructor a 1. We see that our prior selections do result in an apparently reasonable sample of student evaluations, because the proportion of students rating 1 increases as grade and GPA increase. And as class size decreases the proportion of students rating 1 increase.

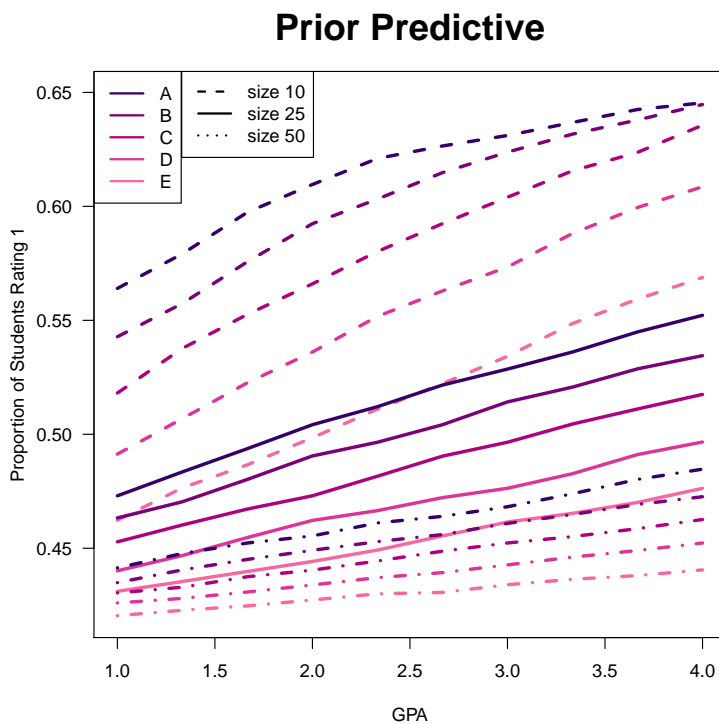


Figure 3.5: The prior predictive distribution shows whether our prior selections result in a reasonable sample of student evaluations.

Next, we create a posterior predictive distribution to see how the addition of the data changes our prior knowledge. With the addition of the data, the prior distributions are updated and called posterior distributions. When creating the prior predictive distribution we used draws from the prior distributions to generate a sample of student evaluations. When creating the posterior predictive distribution we use draws from the posterior distributions to generate a sample of student evaluations.

Figure 3.6 contains a posterior predictive distribution. We can see the influence of the changes in β , particularly the interaction between student grade and GPA. We now see that students with a GPA less than 3.0 and higher grades tend to rate their instructors a 1. However, students with a GPA greater than 3.0 and higher grades are less likely to rate their instructors a 1.

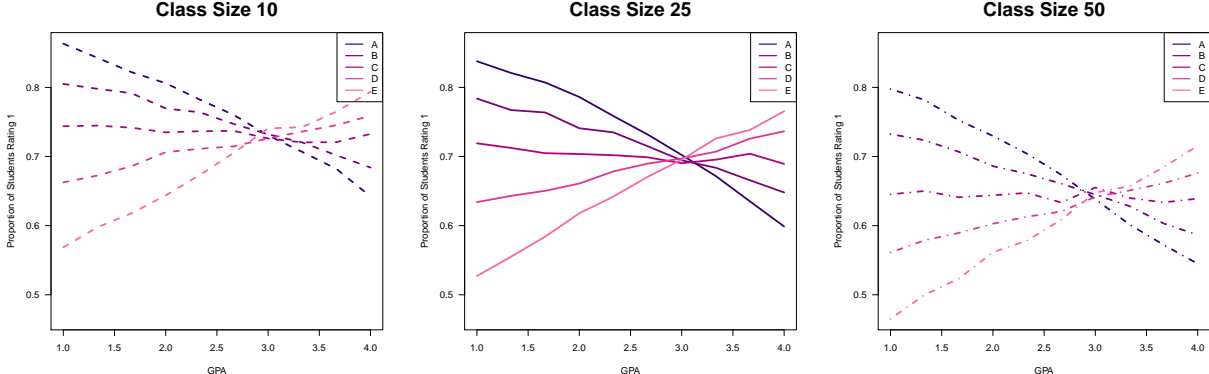


Figure 3.6: Posterior predictive distribution at different levels of the confounders, which shows the effect of incorporating the data into the model.

Conclusion

Brigham Young University students are asked to evaluate each of their instructors on an 8-level Likert scale. However, these evaluations are the student-perceived instructor effectiveness, not the true instructor effectiveness. By using a Bayesian hierarchical probit model we can adjust for the known confounders by including a latent variable that helps to distinguish good instructors from poor instructors.

The posterior distributions for μ_i and σ_i^2 provide an assessment of instructor teaching ability. The posterior distributions for γ_{ij} are used to see the course to course variability within an instructor and to compare instructors teaching the same course.

Overall, this was an effective model in assessing instructor effectiveness by removing the confounding of the student predictors. However, we found that when simplifying the response to a binary variable we lost a lot of information. We were able to distinguish one poor instructor from the rest of the instructors, but we could not distinguish any excellent instructors.

In order to better understand true instructor effectiveness we need to use the 8-level Likert scale, and use a fully ordinal probit model. By extending the response to a multinomial variable we should have more power to distinguish poor, average, and excellent instructors.

ORDINAL HIERARCHICAL PROBIT MODEL OF INSTRUCTOR EFFECTIVENESS

We will again use the data augmentation scheme introduced by [Albert and Chib](#). In the previous model, we used a dichotomous ordinal model, so we only had two categories. In this chapter we will extend this model to a polychotomous setting with more than 2 categories. [Albert and Chib](#) create a system in which it is easy to simulate draws from the posterior distribution; however a Gibbs and Metropolis-Hastings sampler is needed. In this chapter we discuss this extended model in detail and as applied to the student evaluation data.

4.1 CUTPOINT STRUCTURE

In the dichotomous model, we had 2 categories and therefore only one cutpoint. We did not have to estimate this cutpoint because we set it to zero to avoid overparameterization. As we extend the model to the polychotomous settings with $C > 2$ categories, we introducing a vector of cutpoints, $\theta_1, \dots, \theta_{C-1}$, which must be estimated. We define $\boldsymbol{\theta}$ on the z_i so that if $\theta_{c-1} < z_i \leq \theta_c$, we observe y_i in the c^{th} ordered category.

For notational convenience we define $\theta_0 = -\infty$ and $\theta_C = \infty$. Again, the model is overparameterized as we need to estimate $C - 1$ cutpoints, but we only have C categories. As in the dichotomous model we will set $\theta_1 = 0$. [Figure 4.1](#) shows an example of a latent variable distribution for instructor effectiveness with the cutpoints, $\boldsymbol{\theta}$.

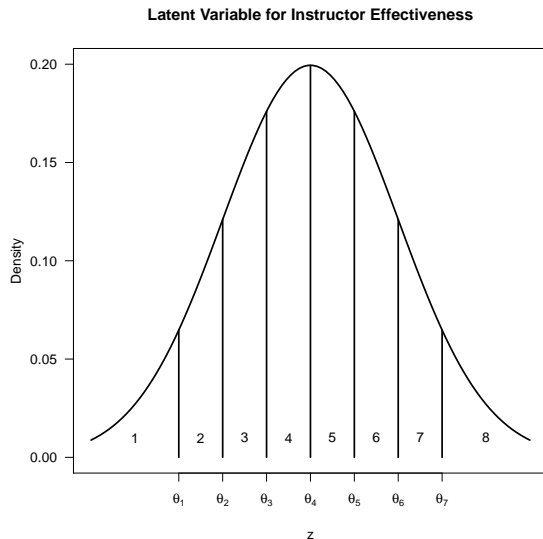


Figure 4.1: Example of latent structure of true instructor effectiveness. We observe the intervals created by the cutpoints, θ .

4.2 ORDINAL HIERARCHICAL PROBIT MODEL WITH LATENT VARIABLE

As we are using a fully ordinal model, the response variable, y_{ijk} , is simply the student rating for the instructor, 1, 2, ..., or 8.

The likelihood function is now multinomial and we define the probabilities for each student to rate their instructor each score, $\mathbf{p}_{ijk} = (p_{(ijk;1)}, \dots, p_{(ijk;C)})$. For example, $p_{(ijk;3)}$ is the probability that student k rates instructor i teaching course j a 3. The multinomial likelihood function is

$$L(\mathbf{p}_{ijk}|y_{ijk}) = \text{Multinomial}(n = 1, p_{(ijk;1)}, \dots, p_{(ijk;C-1)}) = p_{(ijk;y_{ijk})}. \quad (4.1)$$

We still want to account for the differences in instructor and course as well as adjust for the known confounders, so the latent variable structure is the same as in Section 2.3 and is rewritten for convenience,

$$z_{ijk} = \gamma_{ij} + x'_{ijk}\boldsymbol{\beta} + \epsilon_{ijk}, \quad (4.2)$$

where z_{ijk} is student k 's latent instructor effectiveness assessment for instructor i teaching course j , γ_{ij} is the effect of instructor i teaching course j , $\boldsymbol{\beta}$ holds the coefficients for the student predictors, x_{ijk} is a vector of student k 's predictors from course j taught by instructor i , and the errors, ϵ_{ijk} , are distributed $N(0, 1)$.

We will use the probit link function to connect the probability of a student giving a certain rating to the latent variable and confounders,

$$P(ijk; y_{ijk}) = \int_{\theta_{y_{ijk}-1}}^{\theta_{y_{ijk}}} \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta}) dz_{ijk} \quad (4.3)$$

$$= \Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta}). \quad (4.4)$$

From Equation 4.4 the likelihood function is,

$$L(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}) = \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} [\Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta})]. \quad (4.5)$$

Incorporating z , we go from Equation 4.3 to get the likelihood function,

$$L(\mathbf{y}|\boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{\gamma}, \mathbf{z}) = \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} [\varphi(z_{ijk} - \gamma_{ij} - x'_{ijk}\boldsymbol{\beta}) I(\theta_{y_{ijk}-1} \leq z_{ijk} < \theta_{y_{ijk}})]. \quad (4.6)$$

The prior distributions for $\boldsymbol{\beta}$, γ_{ij} , μ_i , and σ_i^2 are the same as outlined in Section 2.3.

Specifically, $\boldsymbol{\beta} \sim N(\mathbf{b}, \mathbf{B})$, with density function

$$\pi(\boldsymbol{\beta}) = \frac{1}{\sqrt{2\pi\mathbf{B}}} e^{-\frac{(\boldsymbol{\beta}-\mathbf{b})^2}{2\mathbf{B}}}, \quad (4.7)$$

$\gamma_{ij} \sim N(\mu_i, \sigma_i^2)$, with density function

$$\pi(\gamma_{ij}) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}}, \quad (4.8)$$

$\mu_i \sim N(m, s^2)$, with density function

$$\pi(\mu_i) = \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(\mu_i-m)^2}{2s^2}}, \quad (4.9)$$

and $\sigma_i^2 \sim IG(a, A)$, with density function

$$\pi(\sigma_i^2) = \frac{1}{\Gamma(a)A^a} (\sigma_i^2)^{-(a+1)} e^{-\frac{A}{\sigma_i^2}}. \quad (4.10)$$

The prior distribution for the cutpoints will be uniform, $\boldsymbol{\theta} \sim \text{Uniform}(\mathbf{u}, \mathbf{v})$,

$$\pi(\boldsymbol{\theta}) = \frac{1}{\mathbf{v} - \mathbf{u}}. \quad (4.11)$$

Hence, the joint posterior distribution is,

$$\pi(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\theta} | \mathbf{y}) \propto \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \left[\pi(\boldsymbol{\beta}) \pi(\mu_i) \pi(\sigma_i^2) \pi(\gamma_{ij}) \pi(\boldsymbol{\theta}) \times \right. \quad (4.12)$$

$$\left. \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) I(\theta_{y_{ijk}-1} \leq z_{ijk} < \theta_{y_{ijk}}) \right]. \quad (4.13)$$

As the complete conditional distributions are available in closed form, a Gibbs Sampler provides draws from the joint posterior distribution. The complete conditional distributions follow:

$$[z_{ijk}] \sim N(\gamma_{ij} + x'_{ijk} \boldsymbol{\beta}, 1), \quad (4.14)$$

truncated on the interval $(\theta_{y_{ijk}-1}, \theta_{y_{ijk}})$,

$$[\boldsymbol{\beta}] \sim N([\mathbf{B}^{-1} + \mathbf{x}'\mathbf{x}]^{-1}[\mathbf{B}^{-1}\mathbf{b} + \mathbf{x}'(\mathbf{z} - \boldsymbol{\gamma})], [\mathbf{B}^{-1} + \mathbf{x}'\mathbf{x}]^{-1}), \quad (4.15)$$

$$[\gamma_{ij}] \sim N\left(\frac{\mu_i + \sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta})}{1 + n_{ij} \sigma_i^2}, \frac{\sigma_i^2}{1 + n_{ij} \sigma_i^2}\right), \quad (4.16)$$

$$[\mu_i] \sim N\left(\frac{\sigma_i^2 m + s^2 \sum_{j=1}^{n_i} \gamma_{ij}}{\sigma_i^2 + n_i s^2}, \frac{s^2 \sigma_i^2}{\sigma_i^2 + n_i s^2}\right), \quad (4.17)$$

$$[\sigma_i^2] \sim IG\left(a + \frac{n_i}{2}, \left[\frac{1}{A} + \frac{\sum_{j=1}^{n_i} (\gamma_{ij} - \mu_i)^2}{2}\right]^{-1}\right), \quad (4.18)$$

and

$$[\theta_c] \sim \text{Uniform}(\max_{y_{ijk}=c-1} z_{ijk}, \min_{y_{ijk}=c} z_{ijk}). \quad (4.19)$$

Sampling from the complete conditional distribution for θ_c results in slow mixing. Instead we use Cowles' algorithm to simulate from the set $\{\mathbf{z}, \boldsymbol{\theta}\}$ (Johnson and Albert 1999). The joint conditional posterior distribution of $(\mathbf{z}, \boldsymbol{\theta})$ can be factored as

$$\pi(\mathbf{z}, \boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}) = \pi(\mathbf{z} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}), \quad (4.20)$$

thus, \mathbf{z} can be sampled from $\pi(\mathbf{z} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}, \boldsymbol{\theta})$, which is the complete conditional distribution in Equation 4.14. And $\boldsymbol{\theta}$ can be sampled from $\pi(\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y})$. Using the likelihood function in Equation 4.5 we see that

$$\begin{aligned} \pi(\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}) = \\ \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} [\Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})], \end{aligned} \quad (4.21)$$

from which we can sample with a Metropolis-Hastings step.

A candidate value, \mathbf{g} , will be sampled from a truncated normal distribution; for $c = 2, \dots, C - 1$, sample $g_c \sim N(\theta_c, \sigma_{MH}^2)$ truncated to the interval (g_{c-1}, θ_{c+1}) . We accept the candidate value, \mathbf{g} , with probability R ,

$$\begin{aligned} R = & \prod_{c=2}^{C-1} \frac{\Phi((\theta_{c+1} - \theta_c)/\sigma_{MH}) - \Phi((g_{c-1} - \theta_c)/\sigma_{MH})}{\Phi((g_{c+1} - g_c)/\sigma_{MH}) - \Phi((\theta_{c-1} - g_c)/\sigma_{MH})} \times \\ & \prod_{i=1}^n \prod_{j=1}^{n_{hi}} \prod_{k=1}^{n_{ij}} \frac{\Phi(g_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(g_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})}{\Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})}. \end{aligned} \quad (4.22)$$

Thus, using both a Gibbs and Metropolis-Hastings sampler, draws can be obtained from the joint posterior distribution.

POLYCHOTOMOUS DATA ANALYSIS

Using the ordinal hierarchical probit model introduced in Section 4.2, we can adjust for the known confounders, and we have the ability to distinguish good, average, and poor instructors. This yields an estimate of instructor effectiveness that can be used to reward or promote the good instructors and advise or remove the poor instructors.

5.1 DATA

The data set we consider is the same data set that was used in the dichotomous hierarchical probit model in Chapter 3 and contains 868 student evaluations collected in a single department over five consecutive years. There are seven instructors in this department collectively teaching 22 courses.

The student predictors included in this analysis are the same as in the dichotomous model, which are class size, student grade in the class, student total GPA, and the interaction between student grade and GPA. Again, student grade in the class was changed to the corresponding grade points used in the calculation of GPA so that grade could be treated as a continuous variable. Each predictor was centered to facilitate mixing and therefore provide better estimates.

Unlike the dichotomous model, which simplified the response to a zero or a one, the response variable is kept as the original 8 categories of the student evaluations.

5.2 PRIOR SELECTION

The prior distributions for μ_i , σ_i^2 , and β are the same as in the dichotomous model, however we are going to make some changes to our prior specifications. Instructor teaching ability,

μ_i , was centered at 1, however we would like to say that an average instructor would have a teaching ability of 0, with poor instructors having a negative teaching ability and excellent instructors having a positive teaching ability. So, we center the prior distribution at zero and give a fairly large variability; specifically

$$\mu_i \sim N(0, 3). \quad (5.1)$$

The variability in the true teaching ability of an instructor within all of the courses taught by that instructor, σ_i^2 , will be kept the same as in the dichotomous model with 97% of the mass between 0 and 1,

$$\sigma_i^2 \sim IG(10, 5). \quad (5.2)$$

The coefficients for the student predictors, $\boldsymbol{\beta}$, are class size, student grade in the class, student GPA, and the interaction between grade and GPA. To make the priors less informative, we will center each β at zero. As discussed previously, when using a probit link function a regression coefficient is interpreted as a z -score. Therefore, this prior suggests that as each predictor increases by one unit the z -score does not change since $\beta = 0$. We will give a fairly large variability of four to each β . Thus, the prior distribution for the regression component is

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{size} \\ \beta_{grade} \\ \beta_{GPA} \\ \beta_{int} \end{bmatrix} \sim N \left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \right). \quad (5.3)$$

No prior specification is needed for $\boldsymbol{\theta}$, the cutpoints. Since the prior is uniformly distributed it contains only constants which are not needed at any point in the Gibbs and Metropolis-Hastings sampler.

5.3 POSTERIOR DISTRIBUTIONS

The Gibbs and Metropolis-Hastings sampler yields posterior draws of μ_i , σ_i^2 , γ_{ij} , $\boldsymbol{\theta}$, z_{ijk} , and $\boldsymbol{\beta}$. 100,000 draws were obtained with 10,000 draws for burn. The acceptance ratio for

θ was 29%. The draws of μ_i and σ_i^2 tell us about each instructor's teaching ability and variability in teaching ability. The draws from γ_{ij} give us a way to compare courses taught by the same instructor and instructors who taught the same course. The draws from θ can be used to estimate the cutpoints of the latent variable distribution, and the draws from z_{ijk} can be used to examine individual students' latent distributions for determining instructor effectiveness. The β 's describe the relationship between the confounders and the probability of receiving a favorable rating.

Posterior Distributions of μ_i and σ_i^2

The posterior distributions for μ_i represent the data-based assessment of the teaching ability of each instructor, and the posterior distributions for σ_i^2 represent the variability in assessed teaching ability of each instructor within all of the courses taught by that instructor.

Figure 5.1 shows the posterior distributions for μ_i and σ_i^2 . When looking at the posterior distributions of μ_i we notice many of the instructors seem to have very similar distributions except for instructor 3086, whose teaching ability seems to be below the rest of the instructors. Also, we notice that instructor 996 seems to have a higher teaching ability than the rest of the instructors. On the other hand, when looking at the posterior distributions for σ_i^2 we find that all the instructors seem to have similar variability in teaching ability.

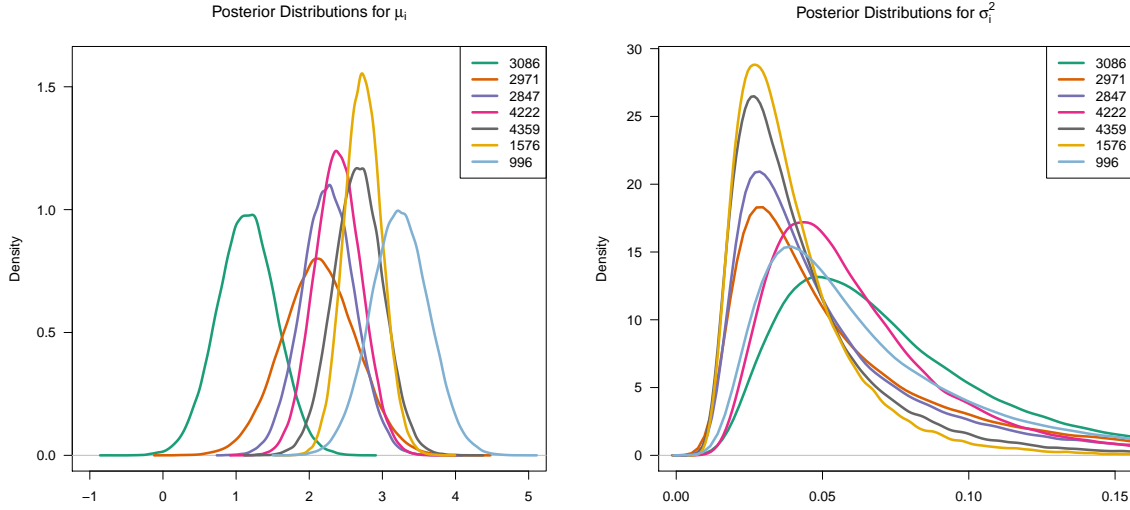


Figure 5.1: Posterior distributions for μ_i , true instructor teaching ability, and σ_i^2 , variability in instructor teaching ability.

Table 5.1 shows the unadjusted mean student ratings and the standard deviation of student ratings for each instructor. Comparing this to the results from the posterior distributions of μ_i and σ_i^2 we find that the posterior distributions seem to follow the order of overall mean scores. Instructor 3086 has very low mean score of 4.09 and is the lowest distribution. Instructors 2847 and 4222 have the same overall mean score of 5.88, yet the posterior distribution of μ_i for instructor 2847 seems to be slightly lower than for instructor 4222. Lastly, we see that instructor 996 has a higher distribution than the rest of the instructors and also has the highest mean overall score of 7.13.

We can formally assess whether there are differences in the instructors' teaching ability, μ_i , by calculating the difference between each draw from each instructor's μ_i and calculating the 95% central credible interval. If the interval contains zero, the instructors' teaching abilities are not significantly different. Table 5.2 shows the 95% credible intervals for the difference between each pair of instructors. Instructor 3086 has a significantly lower teaching ability than all instructors except instructor 2971. Although Figure 5.1 seems to

Table 5.1: Summary statistics of student ratings for each instructor.

Instructor	Overall Mean Score	Standard Deviation
3086	4.09	0.79
2971	5.83	0.34
2847	5.88	0.13
4222	5.88	0.52
4359	6.34	0.18
1576	6.36	0.27
996	7.13	0.25

show some differences in the instructors, this formal test shows that the instructors are generally similar.

Table 5.2: 95% credible intervals for the difference in μ_i for each instructor.

Difference	2.5%	97.5%	Significant	Difference	2.5%	97.5%	Significant
996 - 1576	-0.36	1.36		4359 - 4222	-0.58	1.14	
996 - 4359	-0.39	1.53		4359 - 2847	-0.51	1.33	
996 - 4222	-0.11	1.80		4359 - 2971	-0.62	1.69	
996 - 2847	-0.03	1.99		4359 - 3086	0.55	2.51	*
996 - 2971	-0.11	2.33		4222 - 2847	-0.76	1.02	
996 - 3086	1.03	3.16	*	4222 - 2971	-0.87	1.39	
1576 - 4359	-0.70	0.82		4222 - 3086	0.29	2.21	*
1576 - 4222	-0.39	1.08		2847 - 2971	-1.06	1.31	
1576 - 2847	-0.33	1.29		2847 - 3086	0.10	2.13	*
1576 - 2971	-0.47	1.67		2971 - 3086	-0.24	2.22	
1576 - 3086	0.71	2.48	*				

Posterior Distributions of γ_{ij}

Next, we look at the posterior distributions of γ_{ij} . First, we plot the posterior distributions for an instructor to compare an instructor's teaching ability in different courses. Figure 5.2 shows the lowest rated teacher, instructor 3086, and the highest rated teacher, instructor 996. We can see that there is some variability from course to course for each instructor.

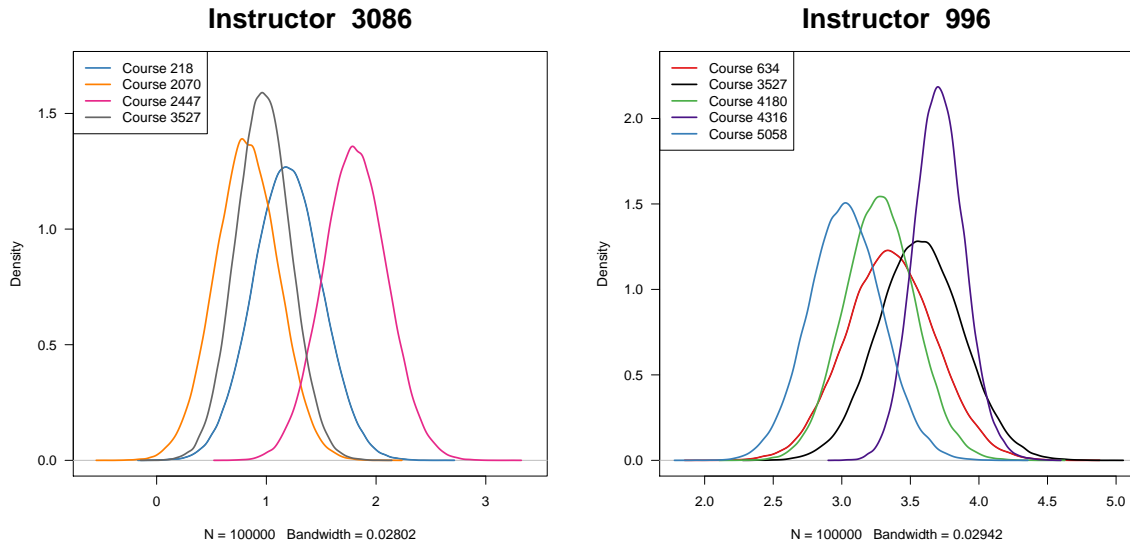


Figure 5.2: Posterior distributions for γ_{ij} by instructor, which allow us to examine the course to course variability within an instructor.

We can also plot the posterior distributions for γ_{ij} for a course and compare instructors who taught the same course. Figure 5.3 shows all the courses that were taught by multiple instructors. In most of the courses the instructors seem to have similar ability to teach. Table 5.3 shows the 95% credible intervals for the difference in γ_{ij} for each instructor who taught the same course. In course 3527, instructor 3086 has a significantly lower ability to teach than the other instructors, and instructor 996 has a significantly higher ability to teach than the other instructors. Also, in course 2070, instructor 3086 has a lower ability to teach than instructor 2847. All the other courses contain non-significant differences in instructor teaching ability; however, in course 634, instructor 996 has a marginally significantly higher teaching ability than instructor 1576.

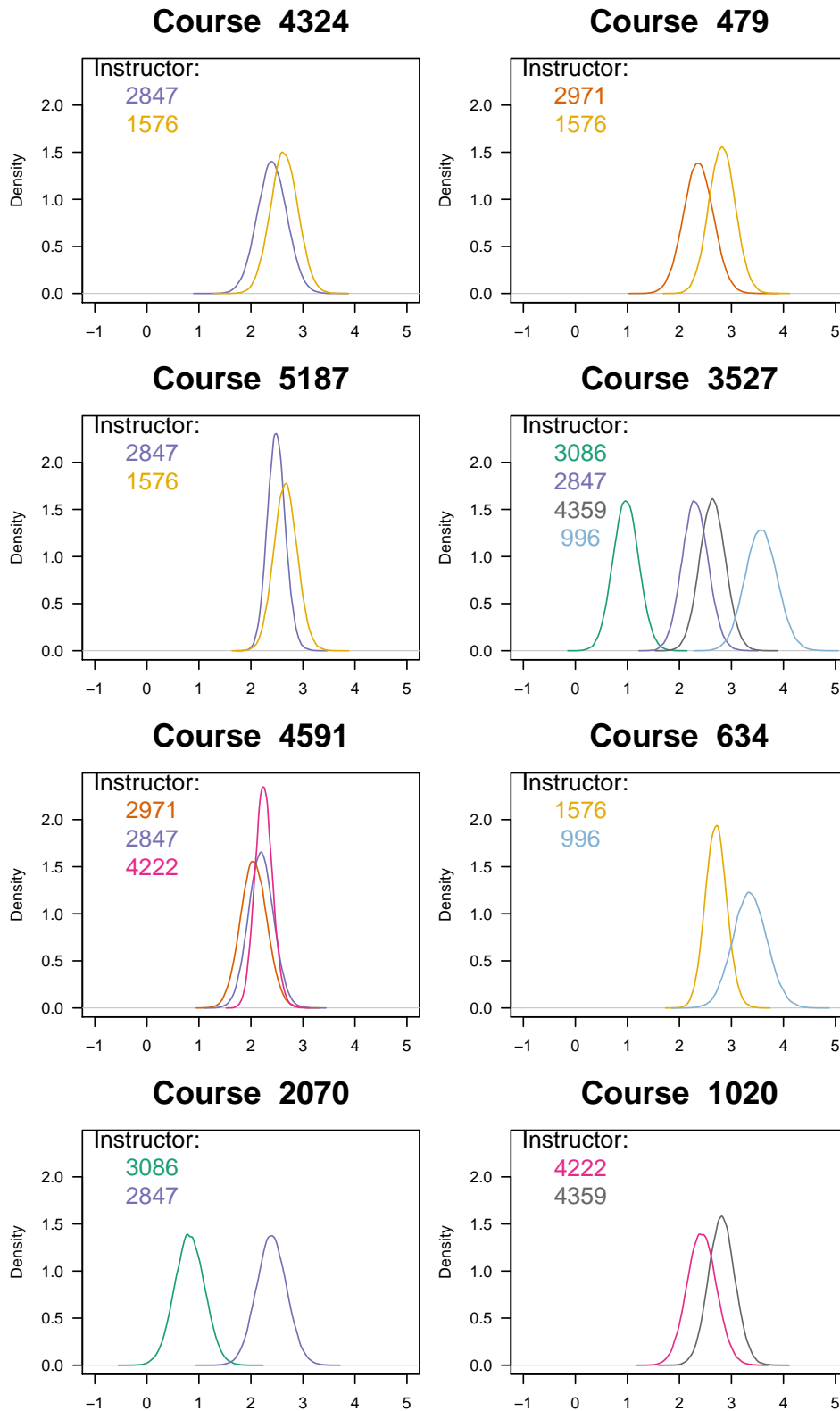


Figure 5.3: Posterior distributions of γ_{ij} for all courses taught by two or more instructors, which enables us to compare instructors who taught the same course.

Table 5.3: 95% credible intervals for the difference in γ_{ij} for each instructor who taught the same course.

Course	Instructor Difference	2.5%	97.5%	Significant
4324	2847 - 1576	-0.92	0.45	
479	2971 - 1576	-1.13	0.21	
5187	2847 - 1576	-0.61	0.26	
3527	3086 - 2847	-1.95	-0.72	*
	3086 - 4359	-2.29	-1.05	*
	3086 - 996	-3.31	-1.90	*
	2847 - 4359	-0.93	0.26	
	2847 - 996	-1.94	-0.60	*
	4359 - 996	-1.62	-0.25	*
4591	2971 - 2847	-0.74	0.46	
	2971 - 4222	-0.69	0.31	
	2847 - 4222	-0.53	0.42	
634	1576 - 996	-1.31	0.001	
2070	3086 - 2847	-2.35	-0.81	*
1020	4222 - 4359	-1.06	0.29	

Posterior Distributions of θ

The posterior distributions for each θ are symmetric and peaked around their means. From the posterior distributions of θ we can obtain estimates of the cutpoints for the latent student assessment of instructor effectiveness. Table 5.4 shows the estimates for the cutpoints, which are the posterior means of each θ_c distribution.

Table 5.4: Posterior means of cutpoints.

	Mean
θ_1	0
θ_2	0.47
θ_3	0.85
θ_4	1.34
θ_5	2.06
θ_6	2.69
θ_7	3.64

Posterior Distributions of z_{ijk}

We also look at the latent variables, z_{ijk} . These latent variables measure the latent student assessment of instructor i 's effectiveness in course j . Recall that if $\theta_{c-1} < z_{ijk} \leq \theta_c$, we observe y_{ijk} in the c^{th} ordered category, or a student rating of c . The cutpoints were estimated in Section 5.3. Figure 5.4 shows the latent variable posterior distributions for each student in course 3527 by instructor. With instructor 3086, we see that most students have a high probability of rating instructor 3086 low, in the range of 1 to 5. On the other hand, all of the students in instructor 996's class have a high probability of rating instructor 996 high, in the range of 5 to 8.

Course 3527

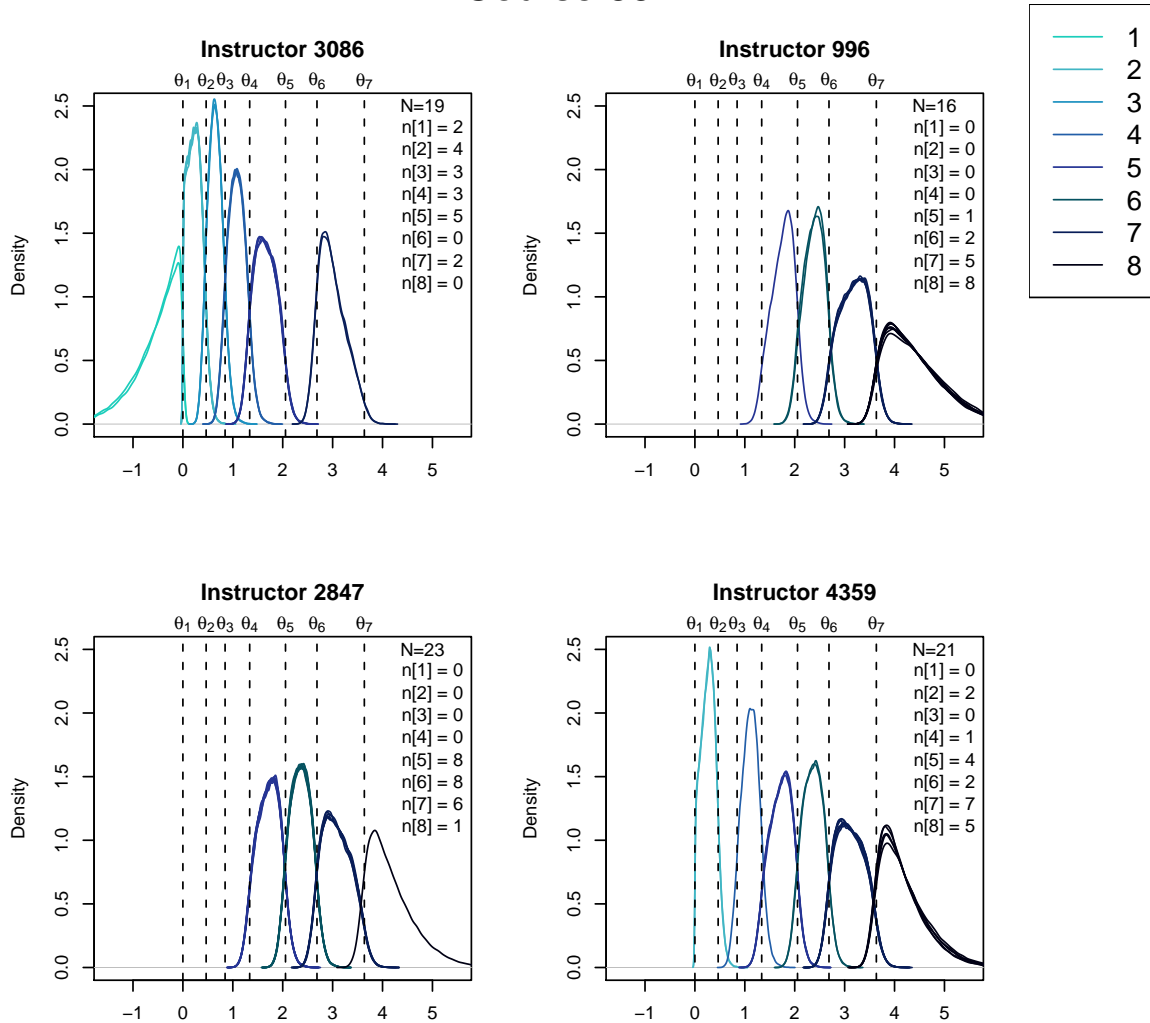


Figure 5.4: Posterior distributions for the latent variable, z_{ijk} , for each instructor who taught course 3527. For each student we are able to examine the probability of the instructor being rated each score.

Posterior Distributions for β

Lastly, we look at the student predictors, β . The posterior distributions for each β are approximately normally distributed. Table 5.5 reports the posterior means for each β as well as the 95% central credible interval (CI). If the 95% CI does not contain zero, we say

that the coefficient is significant. The coefficient for class size, β_{size} , is not significant, thus class size does not seem to affect how a student rates an instructor. However, these data contain a small range of class sizes, only 10–55 students, and we may see a greater influence in class size if we have class sizes in the hundreds. The coefficients for grade and GPA are significant, so grade and GPA affect how a student rates an instructor. The interaction between grade and GPA is not significant, so the coefficients for grade and GPA can be interpreted individually.

As mentioned previously, probit model regression coefficients, $\boldsymbol{\beta}$, can be interpreted as z -scores. As class size increases by 10 students the z -score decreases by -0.008. As student grade increases by one point the z -score increases by 0.2208. As student GPA increases by one point the z -score decreases by -0.3969. As the interaction between student grade and GPA increases by one point the z -score decreases by 0.0549.

Table 5.5: The posterior mean for each β with the 95% central credible interval.

	$E(\cdot \mathbf{y})$	2.5%	97.5%
β_{size}	-0.0008	-0.0105	0.0088
β_{grade}	0.2208	0.1072	0.3345
β_{GPA}	-0.3969	-0.5991	-0.1965
β_{int}	0.0549	-0.1234	0.2323

Predictive Distributions

As in the previous analysis, we examine the prior predictive distribution to ensure our prior selections result in a reasonable sample of student evaluations. We generate 10,000 draws from $\boldsymbol{\beta}$, μ_i , σ_i^2 , and $\boldsymbol{\theta}$. Recall that θ_1 is set to 0 and $\theta_1 < \dots < \theta_7$. So, for each draw of $\boldsymbol{\theta}$ six values were drawn from $\text{Uniform}(\theta_1, 4)$, sorted, and used as θ_2 through θ_7 . Draws from μ_i and σ_i^2 are used to generate 10,000 draws from γ_{ij} . Then, $p_{ijk,y_{ijk}}$ is calculated as,

$$p_{ijk,y_{ijk}} = \Phi(\theta_{y_{ijk}} - \gamma_{ij} - \mathbf{x}'_{ijk}\boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - \mathbf{x}'_{ijk}\boldsymbol{\beta}). \quad (5.4)$$

We can then use those $p_{ijk,y_{ijk}}$ to generate 10,000 draws from the likelihood and see what a sample from that distribution would look like.

Figure 5.5 shows the prior predictive distribution for different levels of the confounders. We have small, medium, and large class sizes, grades A through E, and GPA is kept constant at 3.0, which is close to the average GPA. We examined prior predictive distributions for GPA's from 1.0 to 4.0 and each portrayed the same general pattern we see in the three plots below. The horizontal axis shows the score the instructor received and the vertical axis shows the proportion of students who rated the instructor each score. We see that our prior selections result in a strange phenomenon. There is a high proportion of students rating instructors 1 or 8 regardless of any of the confounders. However, as we recall β is centered at 0, so this prior predictive distribution does not adjust much for the student confounders.

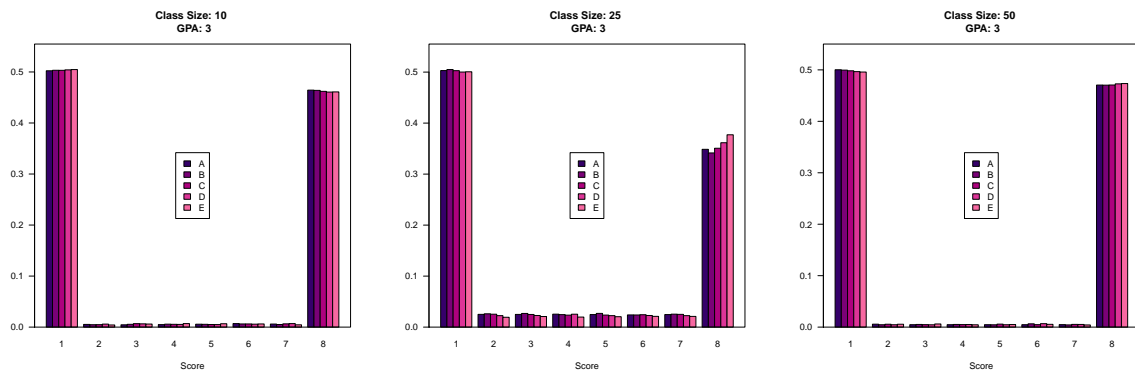


Figure 5.5: The prior predictive distribution shows whether our prior selections result in a reasonable sample of student evaluations.

Next, we create a posterior predictive distribution to see how the addition of the data changes our prior knowledge. When creating the posterior predictive distribution we use the 100,000 draws from the posterior distributions to generate a sample of student evaluations.

Figure 5.6 contains a posterior predictive distribution. We can see the influence of the changes in β . The prior predictive distribution had the strange phenomenon where there was a very high proportion of students rating either a 1 or 8. The posterior predictive

distribution uses more accurate estimates of β , so we get a predictive distribution that is much more reasonable. The lower the student's grade in the course, the more likely the student is to rate the instructor a five or lower. On the other hand, the higher a student's grade in the course, the more likely the student is to rate the instructor a 6 or higher.

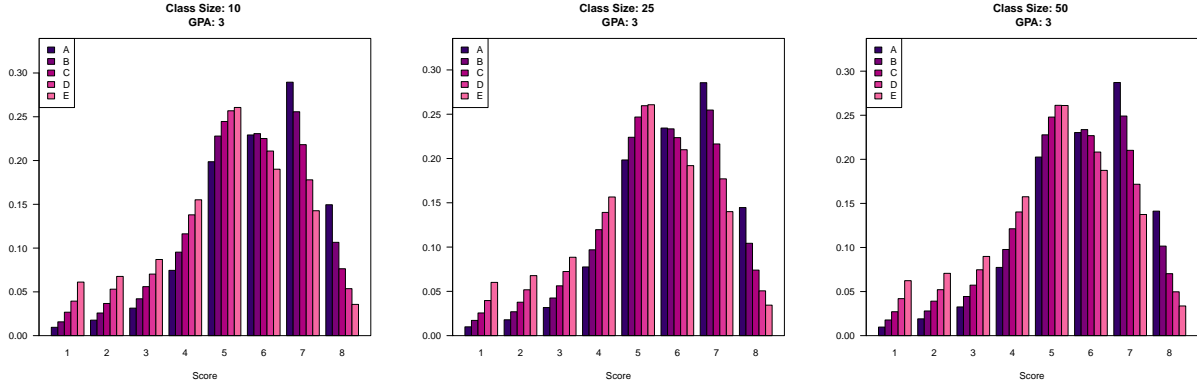


Figure 5.6: Posterior predictive distribution at different levels of the confounders, which shows the effect of incorporating the data into the model.

CONCLUSIONS

Brigham Young University students are asked to evaluate each of their instructors on an 8-level Likert scale. These evaluations are used to help assess instructor effectiveness and to make promotion decisions. However, these evaluations are the student-perceived instructor effectiveness, not the true instructor effectiveness. Using the average of the student evaluations is an inadequate method for assessing instructor effectiveness because it does not account for differences in instructors and courses, and it does not adjust for student confounders. By adjusting for confounding factors we can get a more reasonable assessment of instructor effectiveness.

Using a Bayesian hierarchical probit model we can adjust for the known confounders by including a latent variable that helps to distinguish instructor teaching ability. In the preliminary analysis, to distinguish good instructors from poor instructors, we change the 8-level student ratings to a binary response. Ratings from 1 to 5 get a 0, and ratings from 6 to 8 get a 1. However, we were only able to identify one instructor who seemed to have a low teaching ability. In the final data analysis, we use the full 8-level student ratings and, in some cases, were able to identify poor, average, and exceptional instructors.

We used the posterior distributions of γ_{ij} , plotted by instructor, to see the course to course variability within an instructor. We plotted the posterior distributions of γ_{ij} by course to compare instructors teaching the same course. We found that instructor 3086 did not seem to be as effective as other instructors. We also found that instructor 996 has a higher teaching ability.

The posterior distributions for μ_i and σ_i^2 provide an assessment of instructor teaching ability. Again, we found that the teaching ability of instructor 3086 fell below the teaching

ability of the rest of the instructors. Instructor 996 appeared to have an above average teaching ability, however the difference was not significant.

We looked at the student latent variables, z_{ijk} , to see the probability of a student rating their instructor each score. We found that instructor 3086 had very few students with a high probability of rating a 6–8. On the other hand, all of instructor 996’s students had a high probability of rating a 5–8.

Lastly, we investigated the confounding factors, β . We found that class size and the interaction between grade and GPA did not have a significant effect on the student rating. Grade and GPA did have a significant effect on the student rating.

By changing the response from a binary variable to an ordinal variable, we were able to distinguish poor, average, and exceptional instructors. However, we found that our results followed the unadjusted overall mean scores. Yet, this model gives more insight into instructor effectiveness than the mean scores alone and would be a better tool for making hiring and promotion decisions.

BIBLIOGRAPHY

- Albert, J. H., and Chib, S. (1993), “Bayesian Analysis of Binary and Polychotomous Response Data,” *Journal of the American Statistical Association*, 88, 669–679.
- Carlin, B. P., and Louis, T. A. (2009), *Bayesian Methods for Data Analysis*, Boca Raton, Florida: CRC Press.
- Finney, D. J. (1971), *Probit Analysis*, Great Britain: Cambridge University Press.
- Johnson, V. E. (2008), “Statistical analysis of the National Institutes of Health peer review system,” *Proceedings of the National Academy of Sciences*, 105, 11076–11080.
- Johnson, V. E., and Albert, J. H. (1999), *Ordinal Data Modeling*, New York, New York: Springer.
- Marsh, H. W. (1987), “Students’ evaluations of University teaching: Research findings, methodological issues, and directions for future research,” *International Journal of Educational Research*, 11, 253–388.

APPENDICES

DERIVATION OF COMPLETE CONDITIONAL DISTRIBUTIONS

A.1 HIERARCHICAL PROBIT MODEL

The joint posterior distribution is,

$$\pi(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2 | \mathbf{y}) \propto \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \left[\pi(\boldsymbol{\beta}) \pi(\mu_i) \pi(\sigma_i^2) \pi(\gamma_{ij}) \times \right. \\ \left. \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) [I(z_{ijk} > 0)^{y_{ijk}} + I(z_{ijk} \leq 0)^{1-y_{ijk}}] \right].$$

$$[z_{ijk}] \propto \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) [I(z_{ijk} > 0)^{y_{ijk}} + I(z_{ijk} \leq 0)^{1-y_{ijk}}], \quad (\text{A. 1})$$

which is simply a normal distribution with mean $\gamma_{ij} + x'_{ijk} \boldsymbol{\beta}$ and standard deviation one, truncated on the right at 0 if $y_{ijk} = 0$ and truncated on the left at 0 if $y_{ijk} = 1$.

$$\begin{aligned} [\boldsymbol{\beta}] &\propto \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \left[\pi(\boldsymbol{\beta}) \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) \right] \quad (\text{A. 2}) \\ &= \left[(2\pi)^{-n\beta/2} |\mathbf{B}|^{-1/2} e^{-\frac{1}{2}(\boldsymbol{\beta}-\mathbf{b})' \mathbf{B}^{-1}(\boldsymbol{\beta}-\mathbf{b})} \right] \left[(2\pi)^{-N/2} e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\gamma}-\mathbf{x}\boldsymbol{\beta})'(\mathbf{z}-\boldsymbol{\gamma}-\mathbf{x}\boldsymbol{\beta})} \right] \\ &\propto \exp \left\{ -\frac{1}{2}(\boldsymbol{\beta}-\mathbf{b})' \mathbf{B}^{-1}(\boldsymbol{\beta}-\mathbf{b}) - \frac{1}{2}(\mathbf{z}-\boldsymbol{\gamma}-\mathbf{x}\boldsymbol{\beta})'(\mathbf{z}-\boldsymbol{\gamma}-\mathbf{x}\boldsymbol{\beta}) \right\} \\ &= \exp \left\{ -\frac{1}{2} \left(\boldsymbol{\beta}' \mathbf{B}^{-1} \boldsymbol{\beta} - 2\boldsymbol{\beta}' \mathbf{B}^{-1} \mathbf{b} + \mathbf{b}' \mathbf{B}^{-1} \mathbf{b} + \right. \right. \\ &\quad \left. \left. (\mathbf{z}-\boldsymbol{\gamma})'(\mathbf{z}-\boldsymbol{\gamma}) - 2\boldsymbol{\beta}' \mathbf{x}'(\mathbf{z}-\boldsymbol{\gamma}) + \boldsymbol{\beta}' \mathbf{x}' \mathbf{x} \boldsymbol{\beta} \right) \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \left(\boldsymbol{\beta}' \mathbf{B}^{-1} \boldsymbol{\beta} - 2\boldsymbol{\beta}' \mathbf{B}^{-1} \mathbf{b} - 2\boldsymbol{\beta}' \mathbf{x}'(\mathbf{z}-\boldsymbol{\gamma}) + \boldsymbol{\beta}' \mathbf{x}' \mathbf{x} \boldsymbol{\beta} \right) \right\} \\ &= \exp \left\{ -\frac{1}{2} \left(\boldsymbol{\beta}' [\mathbf{B}^{-1} + \mathbf{x}' \mathbf{x}] \boldsymbol{\beta} - 2\boldsymbol{\beta}' [\mathbf{B}^{-1} \mathbf{b} + \mathbf{x}'(\mathbf{z}-\boldsymbol{\gamma})] \right) \right\} \\ &= \exp \left\{ -\frac{1}{2} [\mathbf{B}^{-1} + \mathbf{x}' \mathbf{x}] \left(\boldsymbol{\beta}' \boldsymbol{\beta} - 2\boldsymbol{\beta}' [\mathbf{B}^{-1} + \mathbf{x}' \mathbf{x}]^{-1} [\mathbf{B}^{-1} \mathbf{b} + \mathbf{x}'(\mathbf{z}-\boldsymbol{\gamma})] \right) \right\} \\ &\sim N([\mathbf{B}^{-1} + \mathbf{x}' \mathbf{x}]^{-1} [\mathbf{B}^{-1} \mathbf{b} + \mathbf{x}'(\mathbf{z}-\boldsymbol{\gamma})], [\mathbf{B}^{-1} + \mathbf{x}' \mathbf{x}]^{-1}) \end{aligned}$$

$$\begin{aligned}
[\gamma_{ij}] &\propto \prod_{k=1}^{n_{ij}} \left[\pi(\gamma_{ij}) \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) \right] & (\text{A. 3}) \\
&= \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\gamma_{ij} - \mu_i)^2}{2\sigma_i^2}} \prod_{k=1}^{n_{ij}} \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})^2}{2}} \right] \\
&\propto e^{-\frac{(\gamma_{ij} - \mu_i)^2}{2\sigma_i^2}} e^{-\sum_{k=1}^{n_{ij}} \frac{(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})^2}{2}} \\
&= \exp \left\{ -\frac{1}{2\sigma_i^2} \left[(\gamma_{ij} - \mu_i)^2 + \sum_{k=1}^{n_{ij}} \sigma_i^2 (z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})^2 \right] \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_i^2} \left[\gamma_{ij}^2 - 2\gamma_{ij}\mu_i + \mu_i^2 + \sum_{k=1}^{n_{ij}} \left(\sigma_i^2 (z_{ijk} - x'_{ijk} \boldsymbol{\beta})^2 - 2\gamma_{ij}\sigma_i^2 (z_{ijk} - x'_{ijk} \boldsymbol{\beta}) + \gamma_{ij}^2 \sigma_i^2 \right) \right] \right\} \\
&\propto \exp \left\{ -\frac{1}{2\sigma_i^2} \left[\gamma_{ij}^2 - 2\gamma_{ij}\mu_i + \sum_{k=1}^{n_{ij}} \left(-2\gamma_{ij}\sigma_i^2 (z_{ijk} - x'_{ijk} \boldsymbol{\beta}) + \gamma_{ij}^2 \sigma_i^2 \right) \right] \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_i^2} \left[\gamma_{ij}^2 - 2\gamma_{ij}\mu_i + -2\gamma_{ij}\sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta}) + \sum_{k=1}^{n_{ij}} (\gamma_{ij}^2 \sigma_i^2) \right] \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_i^2} \left[\gamma_{ij}^2 (1 + n_{ij}\sigma_i^2) - 2\gamma_{ij} \left[\mu_i + \sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta}) \right] \right] \right\} \\
&= \exp \left\{ -\frac{1}{2\sigma_i^2} (1 + n_{ij}\sigma_i^2) \left[\gamma_{ij}^2 - 2\gamma_{ij} \left(\frac{\mu_i + \sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta})}{1 + n_{ij}\sigma_i^2} \right) \right] \right\} \\
&\sim N \left(\frac{\mu_i + \sigma_i^2 \sum_{k=1}^{n_{ij}} (z_{ijk} - x'_{ijk} \boldsymbol{\beta})}{1 + n_{ij}\sigma_i^2}, \frac{\sigma_i^2}{1 + n_{ij}\sigma_i^2} \right)
\end{aligned}$$

$$\begin{aligned}
[\mu_i] &\propto \prod_{j=1}^{n_i} [\pi(\mu_i)\pi(\gamma_{ij})] & (A. 4) \\
&= \frac{1}{\sqrt{2\pi s^2}} e^{-\frac{(\mu_i-m)^2}{2s^2}} \prod_{j=1}^{n_i} \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}} \right] \\
&\propto e^{-\frac{(\mu_i-m)^2}{2s^2}} e^{-\sum_{j=1}^{n_i} \frac{(\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}} \\
&= \exp\left\{ -\frac{1}{2s^2\sigma_i^2} \left[\sigma_i^2(\mu_i-m)^2 + \sum_{j=1}^{n_i} \left(s^2(\gamma_{ij}-\mu_i)^2 \right) \right] \right\} \\
&= \exp\left\{ -\frac{1}{2s^2\sigma_i^2} \left[\sigma_i^2(\mu_i^2 - 2\mu_i m + m^2) + \sum_{j=1}^{n_i} \left(s^2(\gamma_{ij}^2 - 2\mu_i\gamma_{ij} + \mu_i^2) \right) \right] \right\} \\
&\propto \exp\left\{ -\frac{1}{2s^2\sigma_i^2} \left[\mu_i^2\sigma_i^2 - 2\mu_i\sigma_i^2 m - 2\mu_i s^2 \sum_{j=1}^{n_i} \gamma_{ij} + \sum_{j=1}^{n_i} (s^2\mu_i^2) \right] \right\} \\
&= \exp\left\{ -\frac{1}{2s^2\sigma_i^2} \left[\mu_i^2(\sigma_i^2 + n_i s^2) - 2\mu_i \left(\sigma_i^2 m + s^2 \sum_{j=1}^{n_i} \gamma_{ij} \right) \right] \right\} \\
&= \exp\left\{ -\frac{1}{2s^2\sigma_i^2} (\sigma_i^2 + n_i s^2) \left[\mu_i^2 - 2\mu_i \left(\frac{\sigma_i^2 m + s^2 \sum_{j=1}^{n_i} \gamma_{ij}}{\sigma_i^2 + n_i s^2} \right) \right] \right\} \\
&\sim N\left(\frac{\sigma_i^2 m + s^2 \sum_{j=1}^{n_i} \gamma_{ij}}{\sigma_i^2 + n_i s^2}, \frac{s^2 \sigma_i^2}{\sigma_i^2 + n_i s^2} \right)
\end{aligned}$$

$$\begin{aligned}
[\sigma_i^2] &\propto \prod_{j=1}^{n_i} [\pi(\sigma_i^2)\pi(\gamma_{ij})] & (A. 5) \\
&= \frac{1}{\Gamma(a)A^a} (\sigma_i^2)^{-(a+1)} e^{-\frac{1}{A\sigma_i^2}} \prod_{j=1}^{n_i} \left[\frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}} \right] \\
&\propto (\sigma_i^2)^{-(a+1)} e^{-\frac{1}{A\sigma_i^2}} (\sigma_i^2)^{-n_i/2} e^{-\frac{\sum_{j=1}^{n_i} (\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}} \\
&= (\sigma_i^2)^{-(a+n_i/2+1)} e^{-\frac{1}{A\sigma_i^2} - \frac{\sum_{j=1}^{n_i} (\gamma_{ij}-\mu_i)^2}{2\sigma_i^2}} \\
&= (\sigma_i^2)^{-(a+n_i/2+1)} e^{-\frac{1}{\sigma_i^2} \left[\frac{1}{A} + \frac{\sum_{j=1}^{n_i} (\gamma_{ij}-\mu_i)^2}{2} \right]} \\
&\sim IG\left(a + \frac{n_i}{2}, \left[\frac{1}{A} + \frac{\sum_{j=1}^{n_i} (\gamma_{ij}-\mu_i)^2}{2} \right]^{-1} \right)
\end{aligned}$$

A.2 ORDINAL HIERARCHICAL PROBIT MODEL

The joint posterior distribution is,

$$\pi(\boldsymbol{\beta}, \mathbf{z}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\theta} | \mathbf{y}) \propto \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \left[\pi(\boldsymbol{\beta}) \pi(\mu_i) \pi(\sigma_i^2) \pi(\gamma_{ij}) \pi(\boldsymbol{\theta}) \times \right. \\ \left. \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) I(\theta_{y_{ijk}-1} \leq z_{ijk} < \theta_{y_{ijk}}) \right].$$

$$[z_{ijk}] \propto \varphi(z_{ijk} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) I(\theta_{y_{ijk}-1} \leq z_{ijk} < \theta_{y_{ijk}}), \quad (\text{A. 6})$$

which is simply a normal distribution with mean $\gamma_{ij} + x'_{ijk} \boldsymbol{\beta}$ and standard deviation one, truncated to the interval $(\theta_{y_{ijk}-1}, \theta_{y_{ijk}})$.

For the derivation of the complete conditional distributions for $\boldsymbol{\beta}$, γ_{ij} , μ_i , and σ_i^2 see Equations A. 2–A. 5.

$$[\boldsymbol{\theta}] \propto I(\theta_{y_{ijk}-1} \leq z_{ijk} < \theta_{y_{ijk}}) \quad (\text{A. 7}) \\ \sim \text{Uniform}(\max_{y_{hijk}=c-1} z_{ijk}, \min_{y_{hijk}=c} z_{ijk})$$

However, to improve mixing we are drawing from $(\mathbf{z}, \boldsymbol{\theta})$ jointly. The joint conditional posterior distribution of $(\mathbf{z}, \boldsymbol{\theta})$ can be factored as

$$\pi(\mathbf{z}, \boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}) = \pi(\mathbf{z} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}). \quad (\text{A. 8})$$

Thus, \mathbf{z} is sampled from $\pi(\mathbf{z} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}, \boldsymbol{\theta})$ as in Equation A. 6

$\boldsymbol{\theta}$ is sampled from

$$\pi(\boldsymbol{\theta} | \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \mathbf{y}) = \\ \prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} [\Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(\theta_{y_{ijk}-1} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})],$$

which can be sampled from with a Metropolis-Hastings step.

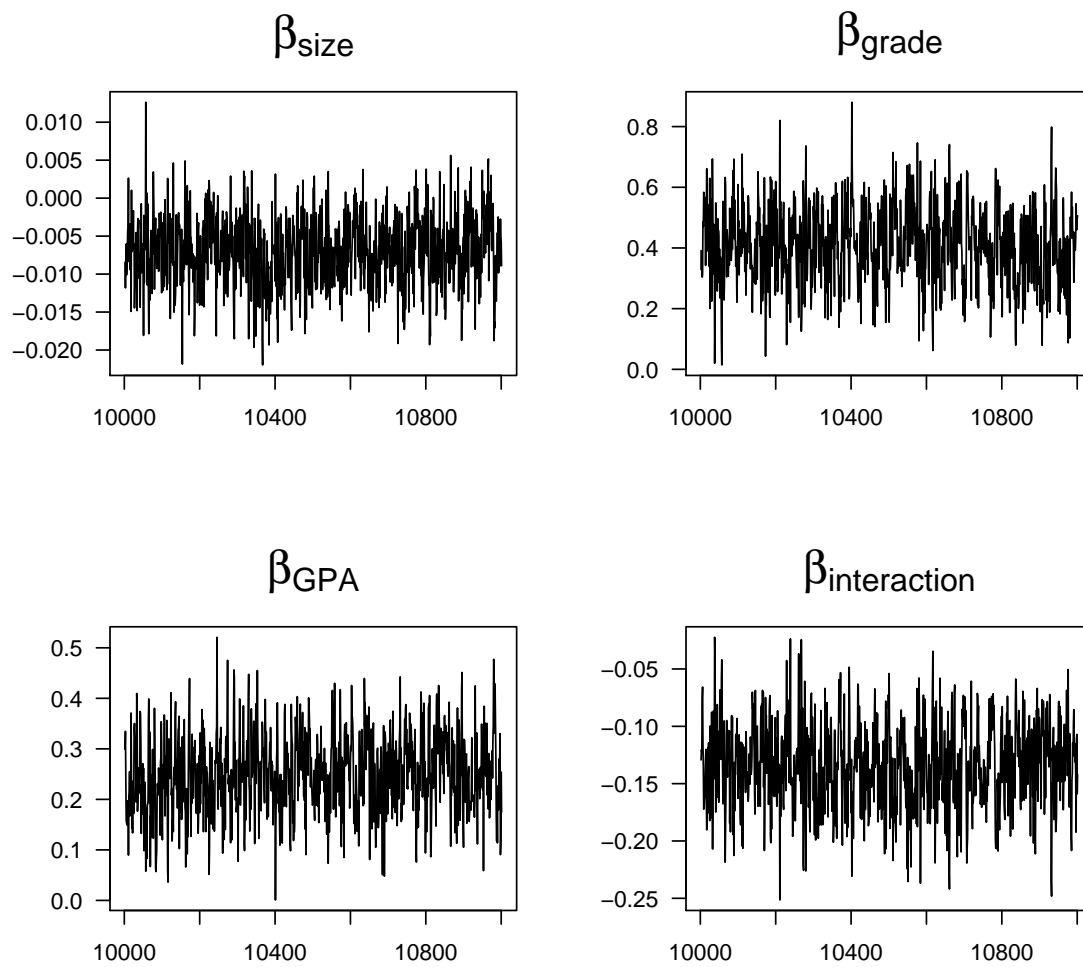
A candidate value, \mathbf{g} , will be sampled from a truncated normal distribution. For $c = 2, \dots, C - 1$, sample $g_c \sim N(\theta_c, \sigma_{MH}^2)$ truncated to the interval (g_{c-1}, θ_{c+1}) . The acceptance ratio is

$$R = \prod_{c=2}^{C-1} \frac{\Phi((\theta_{c+1} - \theta_c)/\sigma_{MH}) - \Phi((g_{c-1} - \theta_c)/\sigma_{MH})}{\Phi((g_{c+1} - g_c)/\sigma_{MH}) - \Phi((\theta_{c-1} - g_c)/\sigma_{MH})} \times$$

$$\prod_{i=1}^n \prod_{j=1}^{n_i} \prod_{k=1}^{n_{ij}} \frac{\Phi(g_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(g_{y_{ijk-1}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})}{\Phi(\theta_{y_{ijk}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta}) - \Phi(\theta_{y_{ijk-1}} - \gamma_{ij} - x'_{ijk} \boldsymbol{\beta})}.$$

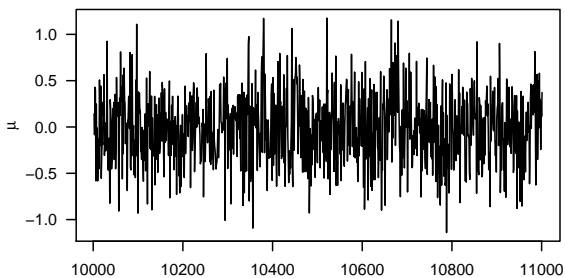
MIXING PLOTS

B.1 MIXING PLOTS FOR BINARY HIERARCHICAL PROBIT MODEL

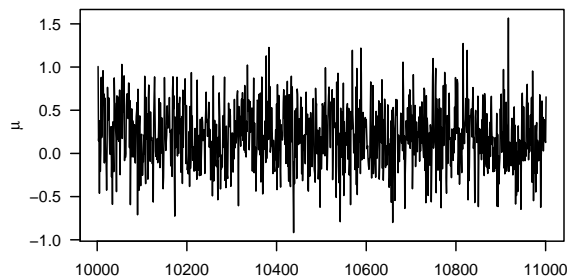
Trace Plots for β 

Trace Plots for μ_i

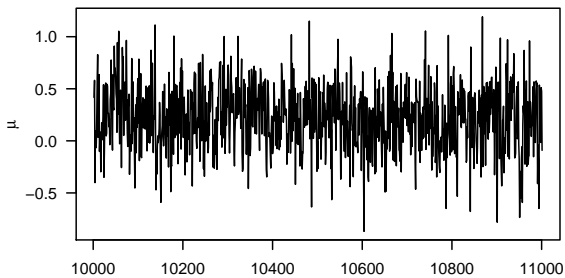
Instructor 2847



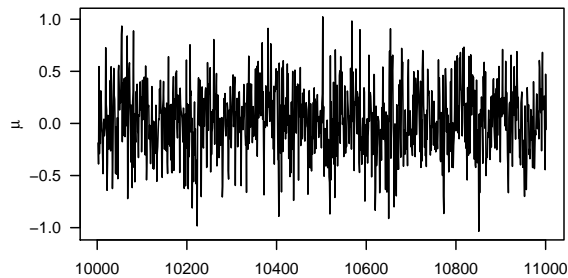
Instructor 1576



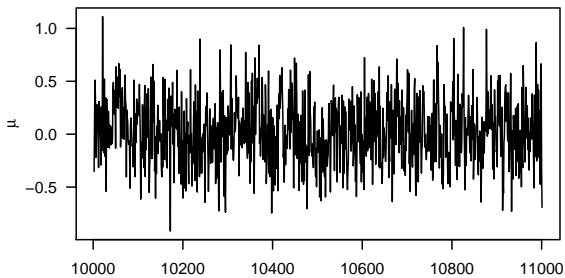
Instructor 996



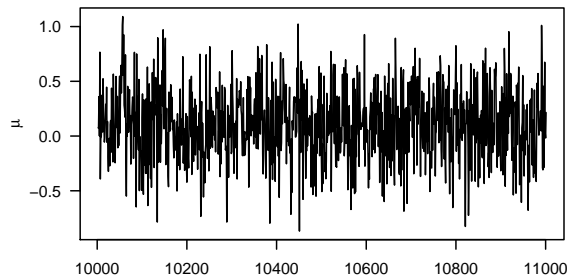
Instructor 2971



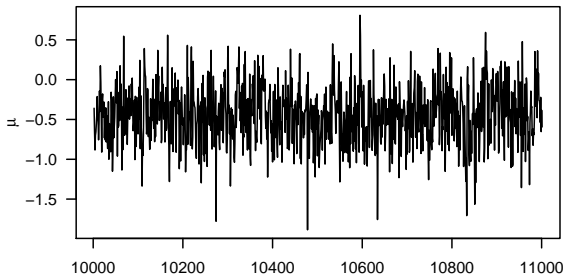
Instructor 4222



Instructor 4359

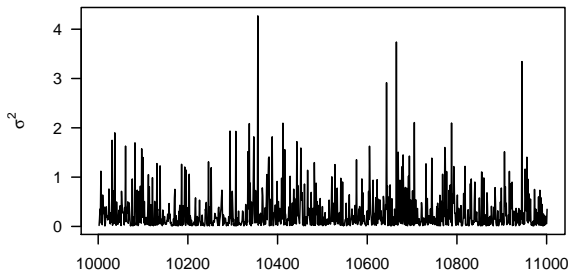


Instructor 3086

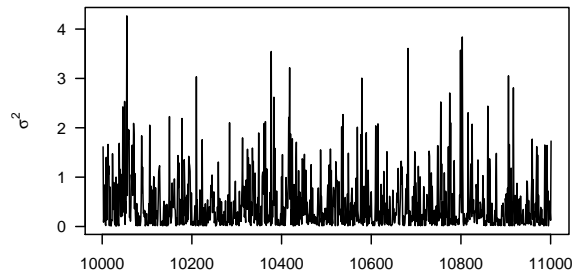


Trace Plots for σ_i^2

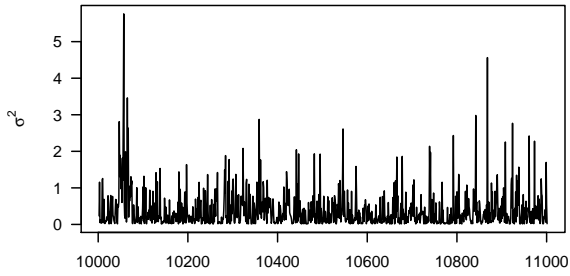
Instructor 2847



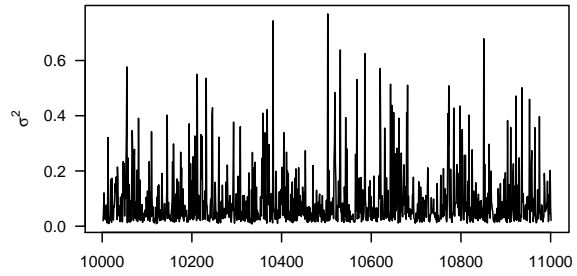
Instructor 1576



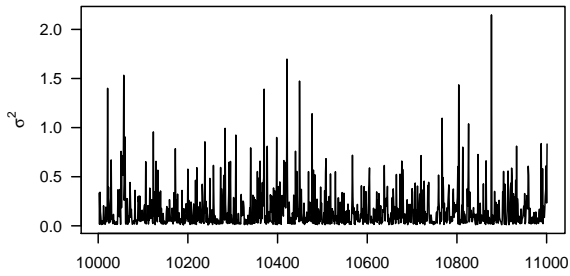
Instructor 996



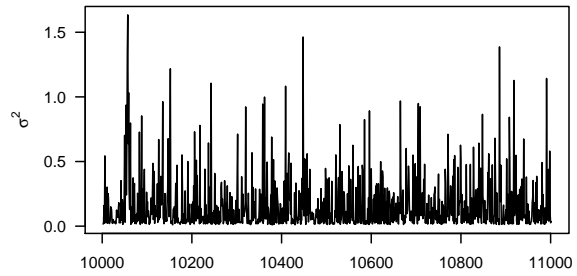
Instructor 2971



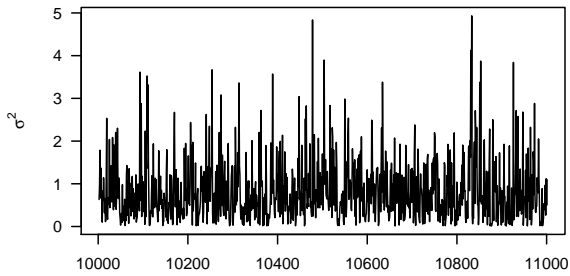
Instructor 4222



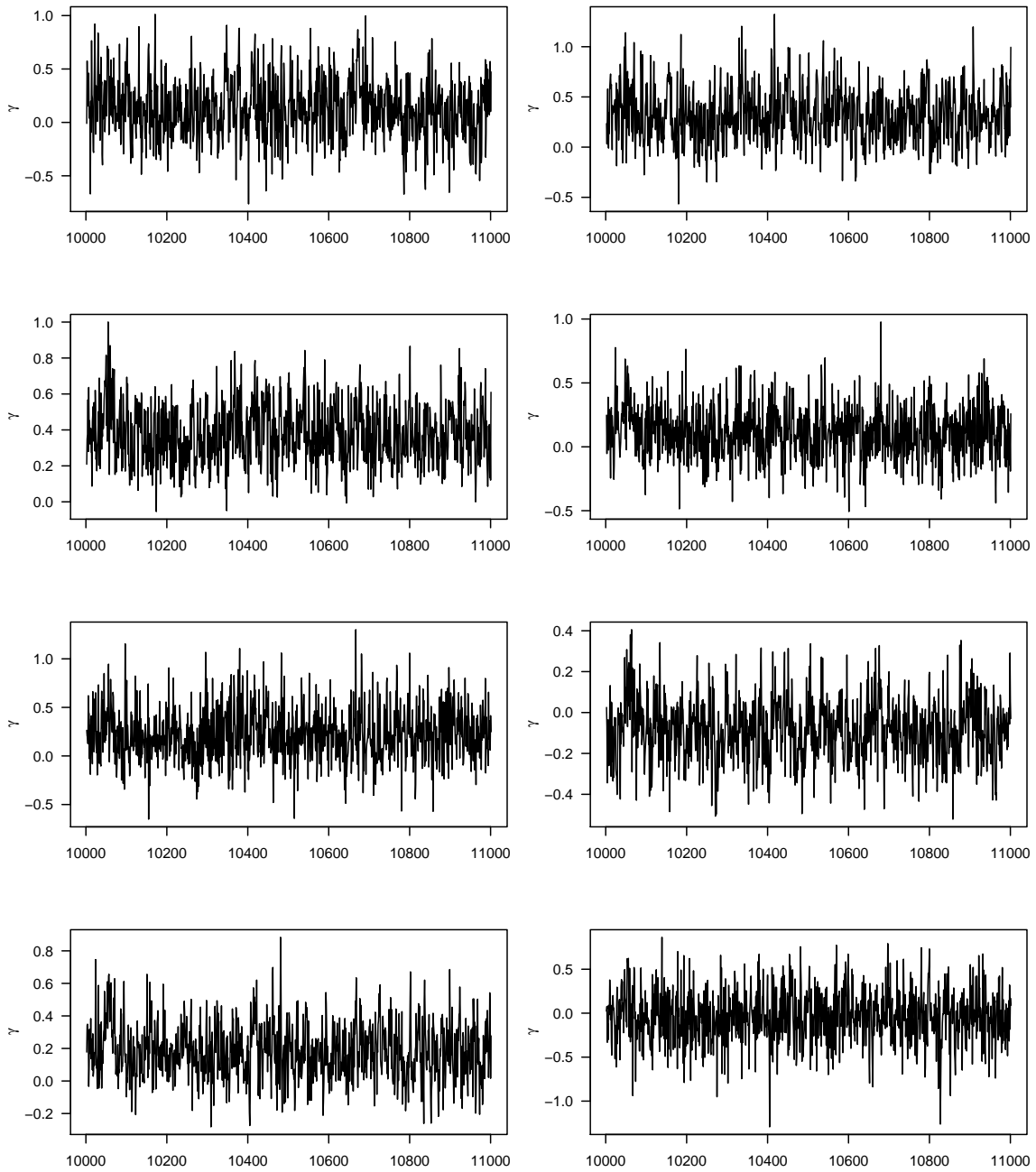
Instructor 4359



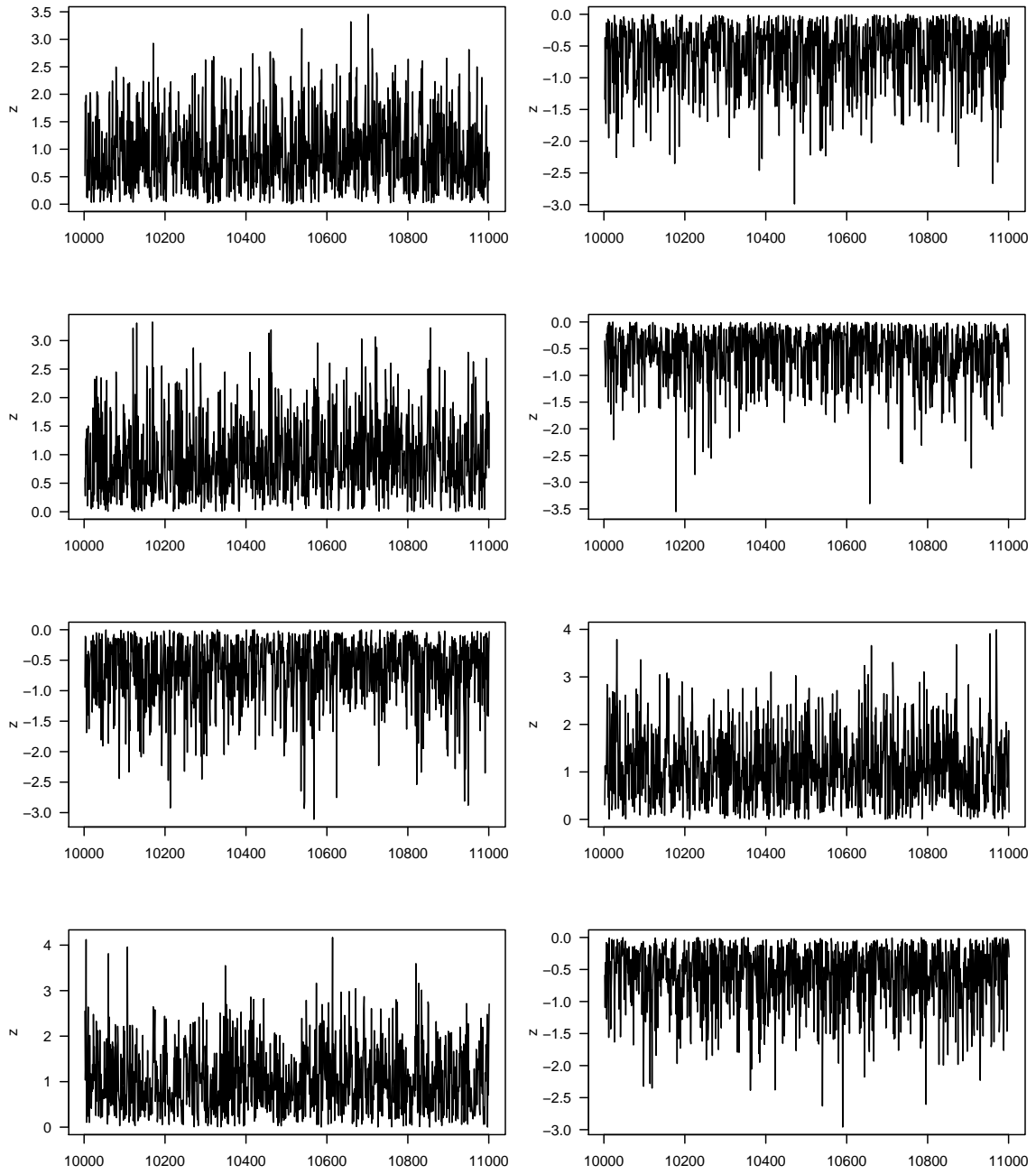
Instructor 3086



Trace Plots for γ_{ij}



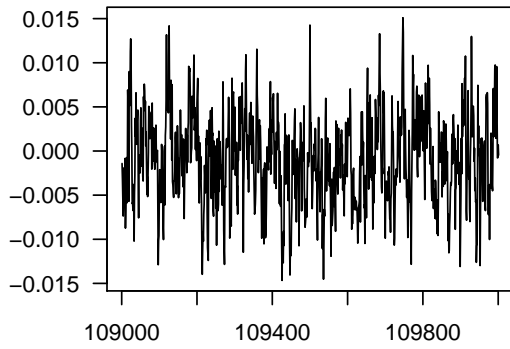
Trace Plots for z_{ijk}



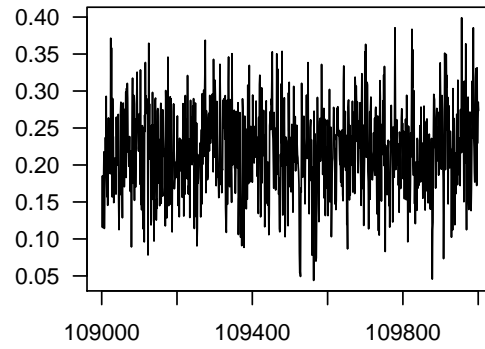
B.2 MIXING PLOTS FOR ORDINAL HIERARCHICAL PROBIT MODEL

Trace Plots for β

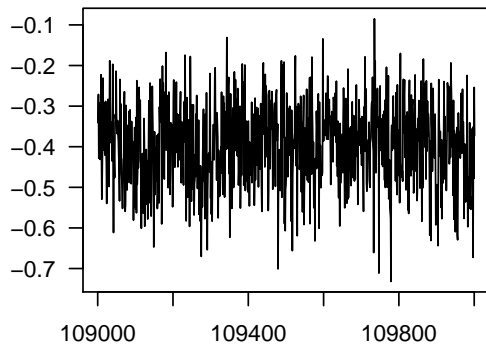
Total_Enrolled



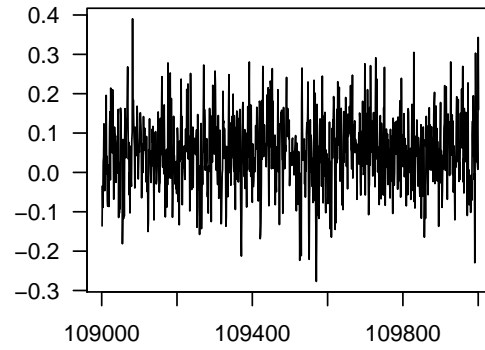
Student_Points



Total_GPA

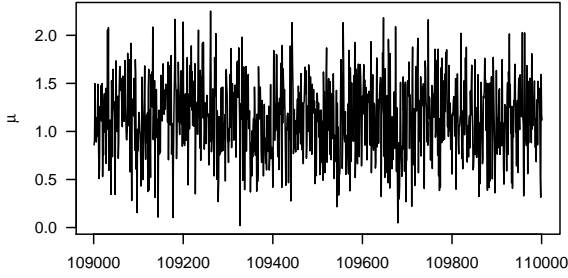


Grade X GPA

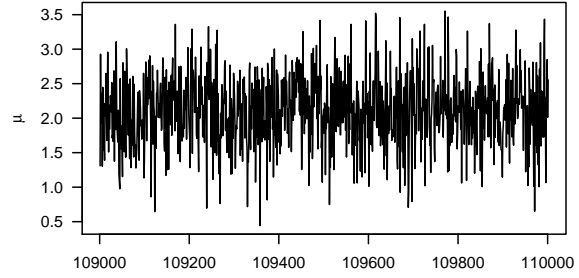


Trace Plots for μ_i

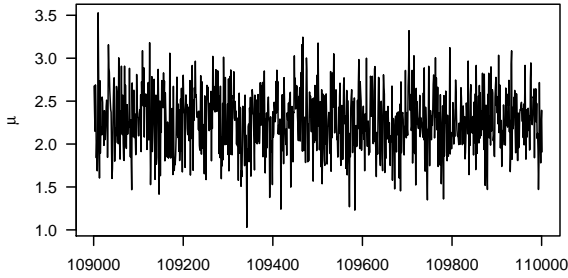
Instructor 3086



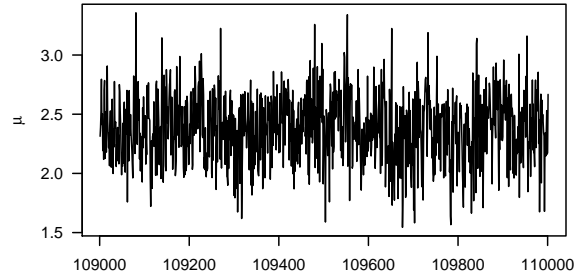
Instructor 2971



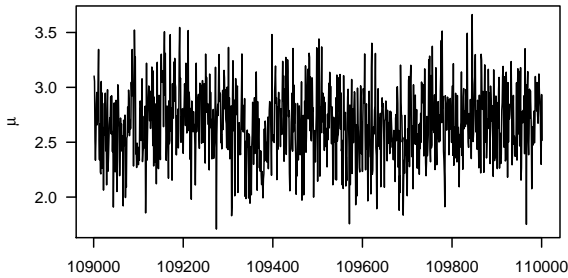
Instructor 2847



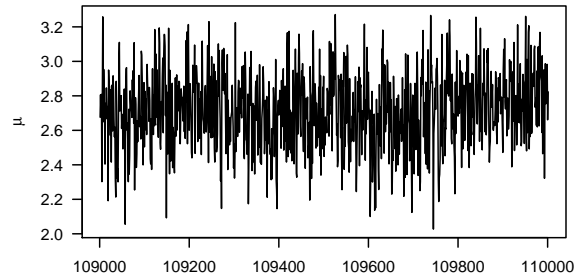
Instructor 4222



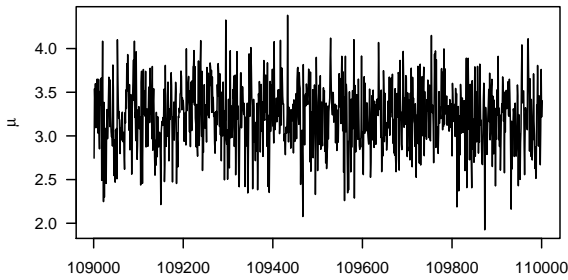
Instructor 4359



Instructor 1576

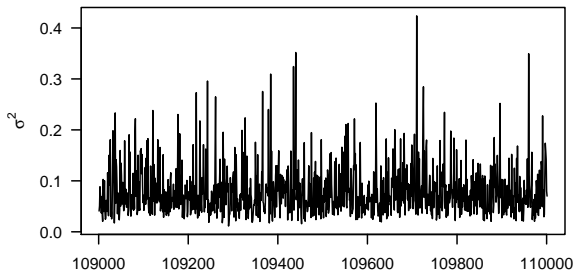


Instructor 996

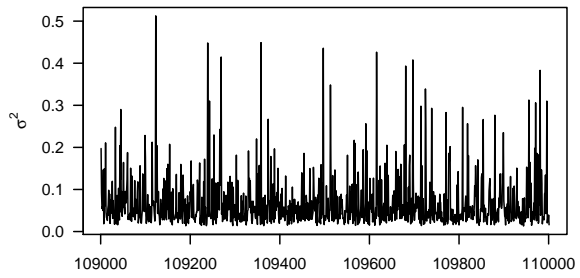


Trace Plots for σ_i^2

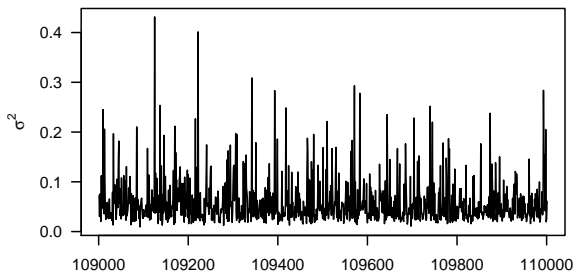
Instructor 3086



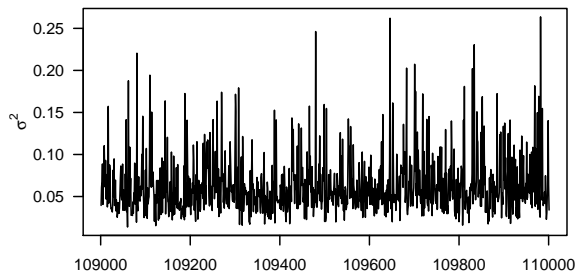
Instructor 2971



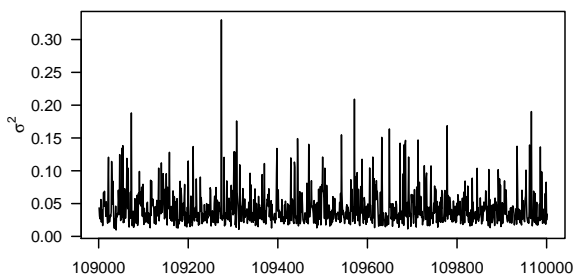
Instructor 2847



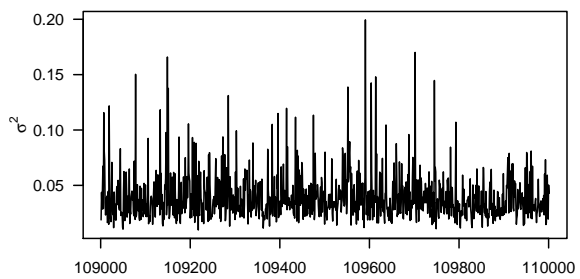
Instructor 4222



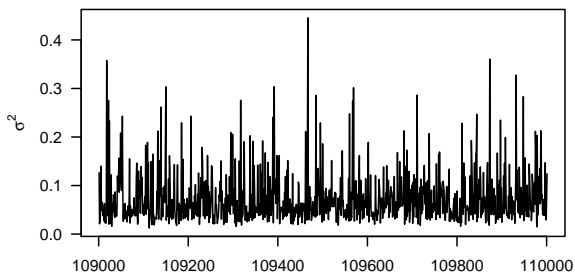
Instructor 4359



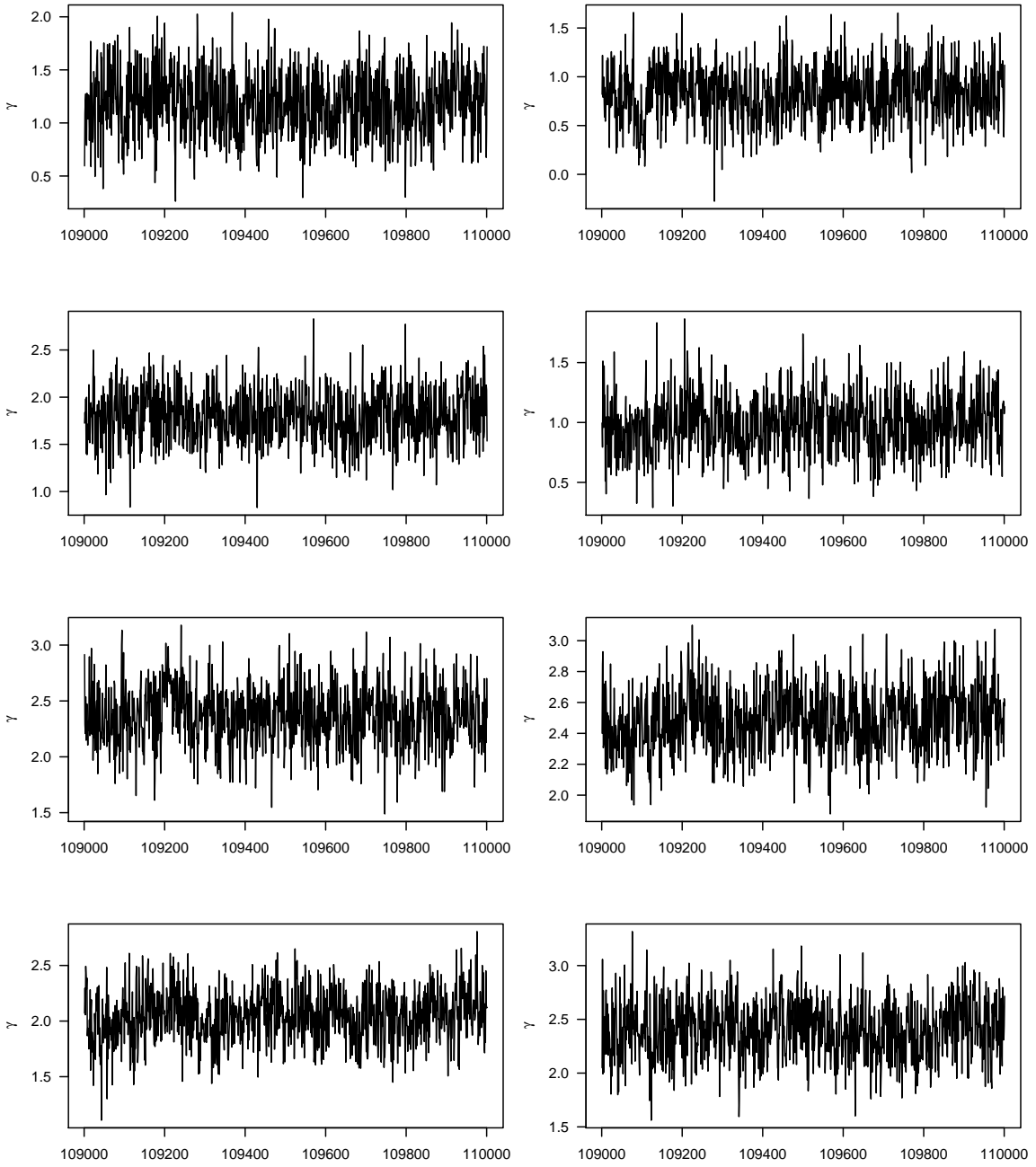
Instructor 1576



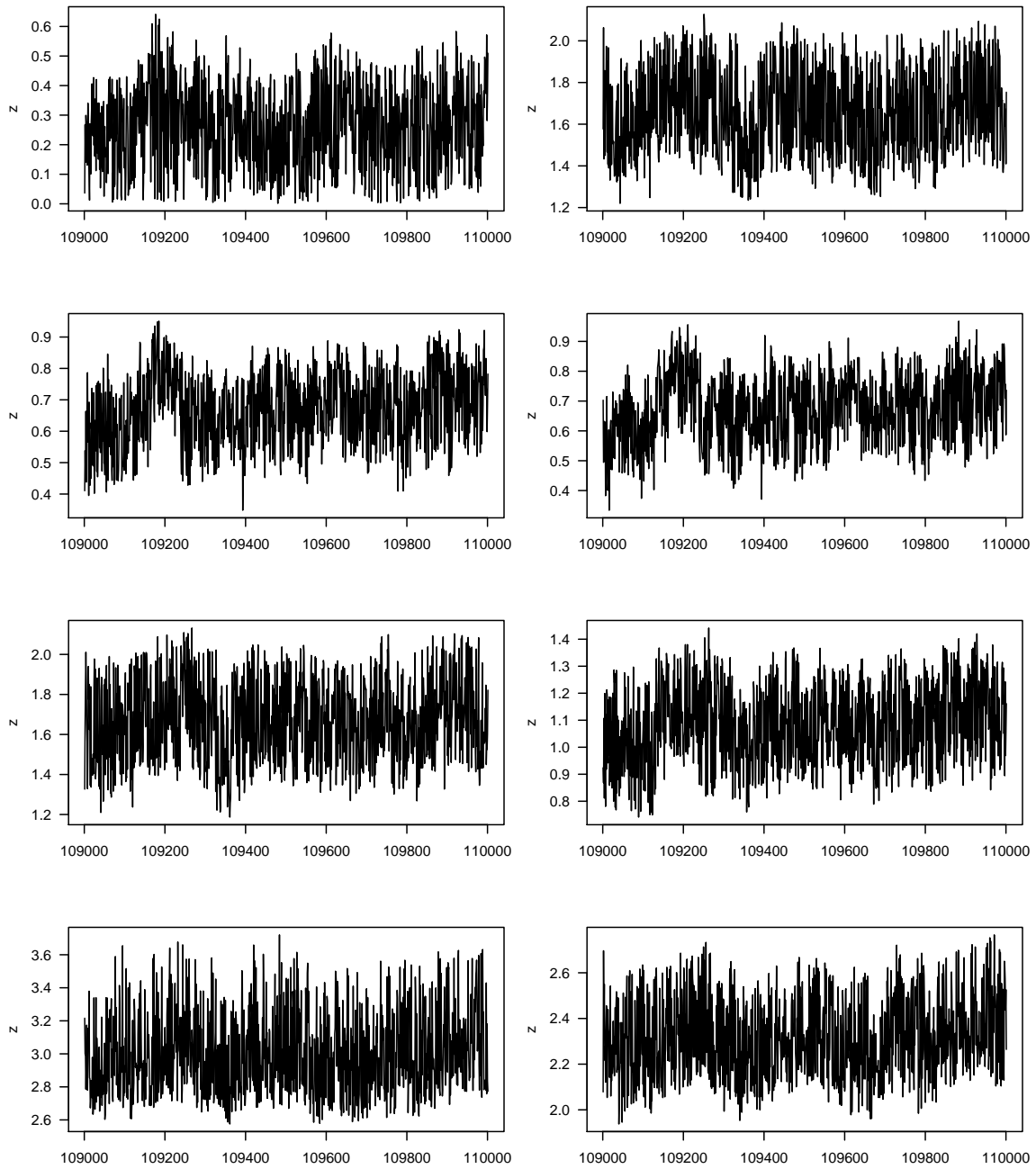
Instructor 996



Trace Plots for γ_{ij}



Trace Plots for Z_{ijk}



R CODE

C.1 CODE FOR HIERARCHICAL PROBIT MODEL

```
setwd("~/Documents/Classes/Project")
evals2<-read.table("~/Documents/Classes/Project/Data_651/evals2.txt",
  header=T)
dim(evals2)
head(evals2)
attach(evals2)
n<-nrow(evals2)
library(xtable)

# rename instructor variable
inst<-bRated_Instructor
# number of instructors
ninst<-length(unique(inst))
# calculate number of courses each instructor taught and average score
ncourse<-inst.mean<-rep(NA,ninst)
for (i in 1:ninst){
  temp<-evals2[inst==unique(inst)[i],]
  ncourse[i]<-length(unique(temp$courseID))
  inst.mean[i]<-mean(ScoOverall_Instructo[inst==unique(inst)[i]])
}
INST<-cbind(unique(inst), ncourse, inst.mean)

grades<-c("A","A-","B+","B","B-","C+","C","C-","D+","D","D-","E")
points<-c(4,3.7,3.4,3,2.7,2.4,2,1.7,1.4,1,.7,0)
# table of grade with grade points
grade.tab<-as.data.frame(cbind(as.character(c("A","A-","B+","B","B-","C
+","C","C-","D+","D","D-","E")), as.numeric(c
(4,3.7,3.4,3,2.7,2.4,2,1.7,1.4,1,.7,0))))
names(grade.tab)<-c("grade","points")
xtable(grade.tab,label="Grade",caption="Caption")

# change letter grades to points
grade.mat<-matrix(NA,ncol=length(grades),nrow=n)
for(i in 1:length(grades)){
  grade.mat[,i]<-ifelse(Student_Grade==grades[i],points[i],100)
}
Student_Points<-apply(grade.mat,1,min)
# Note: there are no D grades or 1 pt

# create unique course variable (course and instructor)
```

```

course.instID<-paste(bcourse_name ,inst ,sep=".")
temp<-unlist(strsplit(unique(course.instID),"\\"))
course2<-temp[seq(1,sum(ncourse)*2,by=2)]
inst2<-temp[seq(2,sum(ncourse)*2,by=2)]
# summary stats for course by instructor
courseBYinst<-as.data.frame(matrix(NA,nrow=sum(ncourse),ncol=12))
courseBYinst[,1]<-inst2
courseBYinst[,2]<-course2
names(courseBYinst)<-c("instructor","course","nevals","credit_hours",
  "grade","age","genderM","ClassStanding","GPA","score","InstStatus",
  "married")
nevals<-rep(NA,sum(ncourse))
for(i in 1:sum(ncourse)){
  temp<-evals2[course.instID==unique(course.instID)[i],]
  courseBYinst[i,3]<-nevals[i]<-nrow(temp)
  courseBYinst[i,4]<-mean(temp$credit_hours)
  courseBYinst[i,5]<-mean(Student_Points[course.instID==unique(course.
    instID)[i]])
  courseBYinst[i,6]<-mean(temp$Age)
  courseBYinst[i,7]<-mean(temp$Gender=="Male")
  courseBYinst[i,8]<-mean(temp$Class_Standing)
  courseBYinst[i,9]<-mean(temp$Total_GPA)
  courseBYinst[i,10]<-mean(temp$ScoOverall_Instructo)
  courseBYinst[i,11]<-mean(temp$bInstructor_status)
  courseBYinst[i,12]<-mean(temp$Marital_Status=="M")
}
courseBYinst

##### center interaction???
# create X matrix with intercept, total enrolled, centered student points,
  centered total GPA, and interaction between points and GPA
#x<-cbind(1,Total_Enrolled,Student_Points-mean(Student_Points),Total_GPA-
  mean(Total_GPA),(Student_Points-mean(Student_Points))*(Total_GPA-mean(
  Total_GPA)) )

# create X matrix with intercept, total enrolled, student points, total
  GPA, and interaction between points and GPA
x<-cbind(Total_Enrolled,Student_Points,Total_GPA,Student_Points*Total_GPA)

# create binary instructor score
  # 0: scores 1-5
  # 1: scores 6-8
y<-ifelse(ScoOverall_Instructo<6,0,1)

#####
library(MASS)
library(MCMCpack)
##### Prior Predictive
set.seed(645)
nn<-10000

# mu_i N(m,s^2)
m<-1
s2<-3

```

```

xx<-seq(-5,5,length=100)
#plot(xx,dnorm(xx,m,sqrt(s2)),type="l")
mu.pred<-rnorm(nn,m,sqrt(s2))
pnorm(0,m,sqrt(s2))

# sigma^2_i IG(a,b)
a<-10
b<-5
xx<-seq(0,5,length=50)
plot(xx,dinvgamma(xx,a,scale=b),type="l")
sig2.pred<-rinvgamma(nn,a,scale=b)
#mean
b/(a-1)
# variance
(b^2)/((a-1)^2*(a-2))

# gamma_ij N(mu_i, sigma2_i)
gamma.pred<-rnorm(nn,mu.pred,sqrt(sig2.pred))
plot(density(gamma.pred))

# beta N(d,g)
nbeta<-4
MLEs<-glm(y~Total_Enrolled+Student_Points+Total_GPA+ Student_Points:
  Total_GPA,family=binomial)
#d<-coef(MLEs)
#g<-diag(summary(MLEs)$coef[,2])
d<-c(-.5,1,2,.5)
g<-diag(c(4,4,4,4))
ginv<-solve(g)
beta.pred<-mvrnorm(nn,mu=d, Sigma=g)

beta.tab<-matrix(NA,nrow=nbeta,ncol=2)
colnames(beta.tab)<-c("mean","prob < 1")
par(mfrow=c(2,3),las=1)
for(i in 1:nbeta){
  plot(density(beta.pred[,i]),main=paste(expression(beta_),i-1))
  beta.tab[i,1]<-mean(beta.pred[,i])
  beta.tab[i,2]<-mean(beta.pred[,i]<1)
}
beta.tab
var((beta.pred))

##### Calculate proportions for prior predicitive distribution
gpa<-seq(1,4,length=10)
pts<-c(4,3,2,1,0)
size<-c(10,25,50)

nnn<-nrow(beta.pred)
prop<-as.data.frame(matrix(NA,ncol=length(gpa),nrow=(length(pts)*length(
  size))))
rownames(prop)<-c("A.10","B.10","C.10","D.10","E.10","A.25","B.25","C
.25","D.25","E.25","A.50","B.50","C.50","D.50","E.50")
colnames(prop)<-round(gpa,2)
for(l in 1:length(size)){

```

```

for(i in 1:length(pts)){
  for(j in 1:length(gpa)){
    p<-pnorm(gamma.pred + c(size[1],pts[i],gpa[j],pts[i]*gpa[j]))**t(
      beta.pred)
    draws<-rbinom(nnn,1,p)
    prop[i+(5*(1-1)),j]<-table(draws)[2]/nnn
  }
}
}

```

```

purple1<-rgb(53,0,106,max=255)
purple2<-rgb(122,1,119,max=255)
purple3<-rgb(174,1,126,max=255)
purple4<-rgb(221,52,151,max=255)
purple5<-rgb(247,104,161,max=255)
purple<-c(purple1,purple2,purple3,purple4,purple5)

```

```

pdf("Plots/prior_pred.pdf")
par(mfrow=c(1,1),las=1)
#main="Prior Predictive Proportion of Favorable Rating"
plot(gpa,prop[1,],type="n",ylim=c(.42,.65),ylab="Proportion of Students
  Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:nrow(prop)){
  lines(gpa,prop[i,],lty=ltype[i],col=color[i],lwd=3)
}
#legend("topright",c("size 10","size 25","size 50"),lty=c(2,1,3),lwd=2)
#legend("topleft",c("A","B","C","D","E"),lty=1,col=c(1:5),lwd=2)
dev.off()

```

```

pdf("Plots/prior_pred_title.pdf")
par(mfrow=c(1,1),las=1)
#main="Prior Predictive Proportion of Favorable Rating"
plot(gpa,prop[1,],type="n",ylim=c(.42,.65),ylab="Proportion of Students
  Rating 1", xlab="GPA",main="Prior Predictive",cex.main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:nrow(prop)){
  lines(gpa,prop[i,],lty=ltype[i],col=color[i],lwd=3)
}
#legend("topright",c("size 10","size 25","size 50"),lty=c(2,1,3),lwd=2)
#legend("topleft",c("A","B","C","D","E"),lty=1,col=c(1:5),lwd=2)
dev.off()

```

```

pdf("Plots/prior_pred_title_doc.pdf")
par(mfrow=c(1,1),las=1)
#main="Prior Predictive Proportion of Favorable Rating"
plot(gpa,prop[1,],type="n",ylim=c(.42,.65),ylab="Proportion of Students
  Rating 1", xlab="GPA",main="Prior Predictive",cex.main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))

```

```

for(i in 1:nrow(prop)){
  lines(gpa,prop[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topleft",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)
legend(1.33,.66,c("size 10","size 25","size 50"),lty=c(2,1,3),lwd=2)
dev.off()

#### instructor by mean scores
courseBYinst[,c(1:3,10)]
ordered.inst<-INST[order(INST[,3]),]

# proportion of 1's for ordered instructors
prop1<-rep(NA,ninst)
for(i in 1:ninst){
  prop1[i]<-mean(y[inst==ordered.inst[i,1]])
}

l1<-as.data.frame(matrix(NA,nrow=ninst,ncol=(max(ordered.inst[,2])+4)))
names(l1)<-c("Instructor","ncourse","inst.mean","prop1","course1","course2",
  "","course3","course4","course5","course6")
sdd<-rep(NA,ninst)
for(i in 1:ninst){
  l1<-courseBYinst$score[courseBYinst[,1]==ordered.inst[i,1]]
  l1[i,(1:(4+length(l1)))]<-c(ordered.inst[i,1:3],prop1[i],l1)
  sdd[i]<-sd(l1)
}

xtable(l1[,-2])

xtable(cbind(l1[,c(1,3,4)],sdd))

#####
library(msm)
set.seed(22)
burn<-1000
nn<-10000
nsim<-nn+burn

comment<-TRUE
if(comment==F){
bnew<-matrix(d,nrow=nsim+1,ncol=nbeta,byrow=T)
# generate intial values for z
low<-ifelse(y==0,-Inf,0)
up<-ifelse(y==0,0,Inf)
znew<-matrix(rtnorm(n,lower=low,upper=up),nrow=nsim+1,ncol=n,byrow=T)
mu<-matrix(rnorm(ninst,m,sqrt(s2)),nrow=nsim+1,ncol=ninst,byrow=T)
sig2<-matrix(rinvgamma(ninst,a,b),nrow=nsim+1,ncol=ninst,byrow=T)
gam_init<-NULL
for(i in 1:ninst){
  gam_init<-c(gam_init,rnorm(ncourse[i],mu[1,i],sqrt(sig2[1,i])))
}
gamma<-matrix(gam_init,nrow=nsim+1,ncol=sum(ncourse),byrow=T)
#cbind(unique(course.instID),gamma[1,])

```

```

for (l in 2:(nsim+1)){
#l<-2
# enlongate gamma
gam2<-NULL
for(i in 1:sum(ncourse)){
  gam2<-c(gam2,rep(gamma[l-1,i],nevals[i])) }

##### update beta
ssinv<-solve(ginv + t(x)%*%x)
bnew[l,]<-mvrnorm(1,mu= ssinv%*%(ginv%*%d + t(x)%*%(znew[l-1,]-gam2)),
  Sigma=ssinv)

##### update z
znew[l,]<-rtnorm(n,mean=(gam2+x%*%bnew[l,]),sd=1,lower=low,upper=up)

##### update mu
# sum gamma by instructor
gamSum<-NULL
for(i in 1:ninst){
  gamSum<-c(gamSum,sum(gamma[l-1,as.numeric(courseBYinst[,1])==unique(
    inst)[i]]))
}

denom<-sig2[l,] + s2*ncourse
num1<-sig2[l,]*m + s2*gamSum
num2<-sig2[l,]*s2
mu[l,]<-rnorm(ninst,num1/denom,sqrt(num2/denom))
##### update sig2
# calculate sum of squares
ssgam<-NULL
for(i in 1:ninst){
  ssgam<-c(ssgam,sum(gamma[l-1,as.numeric(courseBYinst[,1])==unique(
    inst)[i]] + mu[l,i]))
}

siga<-a+ncourse/2
sigb<-1/((1/b)+ssgam^2*.5)
sig2[l,]<-1/rgamma(ninst,siga,scale=sigb)

##### update gamma
# enlongate mu and sig2
mu2<-sig22<-NULL
for(i in 1:ninst){
  mu2<-c(mu2,rep(mu[l,i],ncourse[i]))
  sig22<-c(sig22,rep(sig2[l,i],ncourse[i]))
}
# sum of squares
ssz<-NULL
for(i in 1:sum(ncourse)){
  ssz<-c(ssz,sum(znew[l,bcourse_name==courseBYinst[i,2] &
    bRated_Instructor==courseBYinst[i,1]] - x[bcourse_name==
    courseBYinst[i,2] & bRated_Instructor==courseBYinst[i,1],,]%*%bnew
    [l,]))
}

```

```

    num1<-mu2 + sig22*ssz
    denom<-1 + sig22*nevals
    gamma[l,]<-rnorm(sum(ncourse), num1/denom, sqrt(sig22/denom))
}
write.table(bnew,"beta3_18.txt",row.names=F)
write.table(znew,"z3_18.txt",row.names=F)
write.table(gamma,"gamma3_18.txt",row.names=F)
write.table(mu,"mu3_18.txt",row.names=F)
write.table(sig2,"sig23_18.txt",row.names=F)
}

good<-(-c(1:(burn+1)))

bnew<-read.table("beta3_18.txt",header=T)
znew<-read.table("z3_18.txt",header=T)
gamma<-read.table("gamma3_18.txt",header=T)
mu<-read.table("mu3_18.txt",header=T)
sig2<-read.table("sig23_18.txt",header=T)

bnew_old<-read.table("beta3_17.txt",header=T)
znew_old<-read.table("z3_17.txt",header=T)
gamma_old<-read.table("gamma3_17.txt",header=T)
mu_old<-read.table("mu3_17.txt",header=T)
sig2_old<-read.table("sig23_17.txt",header=T)

good<-(-c(1:(burn+1)))

##### trace plots
# beta
pdf("Plots/trace_beta.pdf",height=5,width=10)
par(mfrow=c(2,2),oma = c(0, 0, 3, 0),las=1)
plot(1:nsim,bnew[-1,1],type="l",xlab="", main =expression(beta[size]),cex.
    main=2,ylab="")
lines(-good,bnew[-good,1],type="l",col="red")
plot(1:nsim,bnew[-1,2],type="l",xlab="", main =expression(beta[grade]),cex.
    .main=2,ylab="")
lines(-good,bnew[-good,2],type="l",col="red")
plot(1:nsim,bnew[-1,3],type="l",xlab="", main =expression(beta[GPA]),cex.
    main=2,ylab="")
lines(-good,bnew[-good,3],type="l",col="red")
plot(1:nsim,bnew[-1,4],type="l",xlab="", main =expression(beta[interaction
    ]),cex.main=2,ylab="")
lines(-good,bnew[-good,4],type="l",col="red")
mtext(expression(paste("Trace Plots for ",beta)),cex=2,outer=T)
dev.off()

# mu_i
pdf("Plots/trace_mu.pdf",height=5,width=10)
par(mfrow=c(2,4),oma = c(0, 0, 3, 0),las=1)
for(i in 1:ninst){
    plot(1:nsim,mu[-1,i],type="l",main=paste("Instructor ",unique(inst)[i])
        ,ylab=expression(mu),xlab="",cex.main=2)
    lines(-good,mu[-good,i],type="l",col="red")
}

```



```

}
mtext(expression(paste("Trace Plots for ",mu[i])),cex=2,outer=T)
dev.off()

# sigma^2_i
# Don't look good
pdf("Plots/trace_sig2.pdf",width=10,height=5)
par(mfrow=c(2,4),oma = c(0, 0, 3, 0),las=1)
for(i in 1:ninst){
  plot(1:nsim,sig2[-1,i],type="l",main=paste("Instructor ",unique(inst)[i
  ]),ylab=expression(paste(sigma^2)),xlab="",cex.main=2)
  lines(-good,sig2[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",sigma[i]^2)),cex=2,outer=T)
dev.off()

# gamma
pdf("Plots/trace_gamma.pdf")
par(mfrow=c(3,3),oma = c(0, 0, 3, 0),las=1)
for(i in 1:9){
  plot(1:nsim,gamma[-1,i],type="l",xlab="",ylab=expression(gamma))
  lines(-good,gamma[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",gamma[ij])),cex=2,outer=T)
dev.off()

# z
pdf("Plots/trace_z.pdf")
par(mfrow=c(3,3),oma = c(0, 0, 3, 0),las=1)
for(i in 1:9){
  plot(1:nsim,znew[-1,i],type="l",xlab="",ylab=expression(z))
  lines(-good,znew[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",z[ijk])),cex=2,outer=T)
dev.off()

##### Posterior Distributions
# posterior distributions for beta
pdf("Plots/beta_post.pdf")
par(mfrow=c(2,2),las=1)
plot(density(bnew[good,1]),main=expression(beta[size]))
plot(density(bnew[good,2]),main=expression(beta[grade]))
plot(density(bnew[good,3]),main=expression(beta[GPA]))
plot(density(bnew[good,4]),main=expression(beta[interaction]))
dev.off()

# posterior dist for interaction
pdf("Plots/beta_interaction.pdf")
par(mfrow=c(1,1),las=1)
plot(density(bnew[good,4]),main=expression(beta[interaction]),cex.main=3)
dev.off()

library("rgl")
# joint dist size and grade

```

```

pdf("Plots/beta_sizegrade.pdf")
par(mfrow=c(1,1),las=1)
persp(kde2d(bnew[good,1],bnew[good,2]),xlab=expression(size),ylab=
  expression(grade),zlab="Density",cex.lab=2)
title(expression(paste(beta[size]," and ",beta[grade],sep="")),cex.main=4)
dev.off()

# joint dist grade and GPA
pdf("Plots/beta_gradeGPA.pdf")
par(mfrow=c(1,1),las=1)
persp(kde2d(bnew[good,2],bnew[good,3]),xlab=expression(grade),ylab=
  expression(GPA),zlab="Density",cex.lab=2)
title(expression(paste(beta[grade]," and ",beta[GPA],sep="")),cex.main=4)
dev.off()

# joint dist size and GPA
pdf("Plots/beta_sizeGPA.pdf")
par(mfrow=c(1,1),las=1)
persp(kde2d(bnew[good,1],bnew[good,3]),xlab=expression(size),ylab=
  expression(GPA),zlab="Density",cex.lab=2)
title(expression(paste(beta[size]," and ",beta[GPA],sep="")),cex.main=4)
dev.off()

# posterior distributions for average z
# average z for each observation
zave<-apply(znew[-good,],2,mean)
pdf("Plots/z_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(zave[y==0]),main=expression(paste("Average ",z[ijk]," for
  each student")),xlim=c(-1,1.8))
lines(density(zave[y==1]))
text(-.4,3,"y=0")
text(1.4,1.5,"y=1")
dev.off()

# posterior distributions for z by instructor and course
pdf("Plots/z_course.pdf")
par(mfrow=c(2,2),oma = c(0, 0, 3, 0),las=1)
tt<-(1:nrow(evals2))[bRated_Instructor==3086 & bcourse_name==3527]
plot(density(znew[-good,tt[1]]),ylim=c(0,1.2),xlim=c(-3,3),main="
  Instructor 3086",xlab="")
for(j in 1:length(tt)){
  lines(density(znew[-good,tt[j]]))
}
abline(v=0)
text(-1,1,"0",cex=2)
text(1,1,"1",cex=2)

tt<-(1:nrow(evals2))[bRated_Instructor==996 & bcourse_name==3527]
plot(density(znew[-good,tt[1]]),ylim=c(0,1.2),xlim=c(-3,3),main="
  Instructor 996",xlab="")
for(j in 1:length(tt)){
  lines(density(znew[-good,tt[j]]))
}

```

```

abline(v=0)
text(-1,1,"0",cex=2)
text(1,1,"1",cex=2)
tt<-(1:nrow(evals2))[bRated_Instructor==2847 & bcourse_name==3527]
plot(density(znew[-good,tt[1]]),ylim=c(0,1.2),xlim=c(-3,3),main="
  Instructor 2847",xlab="")
for(j in 1:length(tt)){
  lines(density(znew[-good,tt[j]]))
}
abline(v=0)
text(-1,1,"0",cex=2)
text(1,1,"1",cex=2)
tt<-(1:nrow(evals2))[bRated_Instructor==4359 & bcourse_name==3527]
plot(density(znew[-good,tt[1]]),ylim=c(0,1.2),xlim=c(-3,3),main="
  Instructor 4359",xlab="")
for(j in 1:length(tt)){
  lines(density(znew[-good,tt[j]]))
}
abline(v=0)
text(-1,1,"0",cex=2)
text(1,1,"1",cex=2)
mtext("Course 3527",cex=1.5,outer=T)
dev.off()

# posterior distribution for gamma
pdf("Plots/gamma_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(gamma[good,1]),xlim=c(-2.5,2.5),ylim=c(0,2))
for(i in 2:sum(ncourse)){
  lines(density(gamma[good,i]),col=i)
}
dev.off()

# 7 colors for courses
c1<-rgb(228,26,28,max=255)
c2<-rgb(55,126,184,max=255)
c3<-rgb(77,175,74,max=255)
c4<-rgb(102,102,102,max=255)
c5<-rgb(255,127,0,max=255)
c6<-rgb(231,41,138,max=255)
c7<-rgb(74,20,134,max=255)
c8<-"black"
c.col<-c(c1,c2,c3,c4,c5,c6,c7)
c.col1<-c(c1,c8,c3,c7,c2)
c.col2<-c(c2,c5,c6,c4)

# gamma for each instructor
pdf("Plots/gammaInst_post.pdf",height=5,width=10)
par(mfrow=c(2,4),las=1)
for(j in 1:ninst){
  temp<-gamma[good,courseBYinst[,1]==unique(inst)[j]]
  #par(mfrow=c(1,1))
  plot(density(temp[,1]),xlim=c(-2.5,2.5),ylim=c(0,2.7),main=paste("
    Instructor ",unique(inst)[j]),type="n")
}

```

```

    text(-1,2.7,"Course:")
    for (i in 1:ncol(temp)){
      lines(density(temp[,i]),col=c.col[i])
      text(-1.5,(2.7-(.2*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
        ],2][i],col=c.col[i])
    }
  }
}
dev.off()

# gamma for instructor 996
pdf("Plots/gammaInst_postBest.pdf")
par(mfrow=c(1,1),las=1)
ii<-996
temp<-gamma[good,courseBYinst[,1]==ii]
#par(mfrow=c(1,1))
plot(density(temp[,1]),xlim=c(-1,1.8),ylim=c(0,2.6),main=paste("
  Instructor ",ii),cex.main=2)
#text(-1.3,2,"Course:",cex=1.5)
for (i in 1:ncol(temp)){
  lines(density(temp[,i]),col=c.col1[i],lwd=2)
  #text(-1.3,(2-(.2*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
    ],2][i],col=c.col[i],cex=1.5)
}
legend("topleft",paste("Course",courseBYinst[courseBYinst[,1]==ii,2]),col=
  c.col1,lty=1,lwd=2)
dev.off()

# gamma for instructor 6
pdf("Plots/gammaInst_post6.pdf")
par(mfrow=c(1,1),las=1)
for(j in 6){
  temp<-gamma[good,courseBYinst[,1]==unique(inst)[j]]
  #par(mfrow=c(1,1))
  plot(density(temp[,1]),xlim=c(-1.5,1.5),ylim=c(0,2),main=paste("
    Instructor ",unique(inst)[j]),cex.main=2)
  #text(-1.3,2,"Course:",cex=1.5)
  for (i in 1:ncol(temp)){
    lines(density(temp[,i]),col=c.col1[i],lwd=2)
    #text(-1.3,(2-(.2*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
      ],2][i],col=c.col[i],cex=1.5)
  }
}
legend("topleft",paste("Course",courseBYinst[courseBYinst[,1]==unique(inst
  ) [6],2]),col=c.col1,lty=1,lwd=2)
dev.off()

# gamma for instructor 7
pdf("Plots/gammaInst_post7.pdf")
par(mfrow=c(1,1),las=1)
for(j in 7){
  temp<-gamma[good,courseBYinst[,1]==unique(inst)[j]]
  #par(mfrow=c(1,1))

```

```

plot(density(temp[,1]),xlim=c(-2.5,1),ylim=c(0,1.5),main=paste("
  Instructor ",unique(inst)[j]),cex.main=2)
#text(-2,1.5,"Course:",cex=1.5)
for (i in 1:ncol(temp)){
  lines(density(temp[,i]),col=c.col2[i],lwd=2)
  #text(-2,(1.5-(.15*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
    ],2][i],col=c.col[i],cex=1.5)
}
}
legend("topleft",paste("Course",courseBYinst[courseBYinst[,1]==unique(inst
  )][7],2)),col=c.col2,lty=1,lwd=2)
dev.off()

# colors for 7 instructors
inst1<-rgb(27,158,119,max=255)
inst2<-rgb(217,95,2,max=255)
inst3<-rgb(117,112,179,max=255)
inst4<-rgb(231,41,138,max=255)
inst5<-rgb(102,102,102,max=255)
inst6<-rgb(230,171,2,max=255)
inst7<-rgb(128,177,211,max=255)
inst.col<-c(inst1,inst2,inst3,inst4,inst5,inst6,inst7)

# gamma for each course
# calculate number of instructors teaching each course
ninstBYcourse<-unique(bcourse_name)
for(j in 1:length(unique(bcourse_name))){
  ninstBYcourse[j]<-sum(courseBYinst[,2]==unique(bcourse_name)[j])
}
sum(ninstBYcourse>1)
multInstCourses<-unique(bcourse_name)[ninstBYcourse>1]

pdf("Plots/gammaCourse_post.pdf",height=5,width=10)
par(mfrow=c(2,4),las=1)
for(j in 1:sum(ninstBYcourse>1)){
  temp<-gamma[good,courseBYinst[,2]==multInstCourses[j]]
  #par(mfrow=c(1,1))
  plot(density(temp[,1]),xlim=c(-2.5,2.5),ylim=c(0,2.65),main=paste("
    Course ", multInstCourses[j]),type="n",cex.main=2)
  text(-1.4,2.5,"Instructor:",cex=1.5)
  temp.inst<-courseBYinst[courseBYinst[,2]==multInstCourses[j],1]
  for (i in 1:ncol(temp)){
    lines(density(temp[,i]),col=inst.col[unique(inst)[order(INST[,3])]==
      temp.inst[i]])
    text(-1.8,(2.5-(.4*i)),temp.inst[i],col=inst.col[unique(inst)[order(
      INST[,3])]==temp.inst[i]],cex=1.5)
  }
}
}
dev.off()

pdf("Plots/gammaCourse_post2.pdf",height=10,width=6)
par(mfrow=c(4,2),las=1,mar=c(2,3.85,4,1))
for(j in 1:sum(ninstBYcourse>1)){
  temp<-gamma[good,courseBYinst[,2]==multInstCourses[j]]

```

```

#par(mfrow=c(1,1))
plot(density(temp[,1]),xlim=c(-2.5,2.5),ylim=c(0,2.65),main=paste("
  Course ", multInstCourses[j]),type="n",cex.main=2)
text(-1.4,2.5,"Instructor:",cex=1.5)
temp.inst<-courseBYinst[courseBYinst[,2]==multInstCourses[j],1]
for (i in 1:ncol(temp)){
  lines(density(temp[,i]),col=inst.col[unique(inst)[order(INST[,3])]==
    temp.inst[i]])
  text(-1.8,(2.5-(.4*i)),temp.inst[i],col=inst.col[unique(inst)[order(
    INST[,3])]==temp.inst[i]],cex=1.5)
}
}
dev.off()

layout(matrix(c(2,2,2,2), 2, 2, byrow = TRUE))
## show the regions that have been allocated to each plot
layout.show(2)

# posterior distributions for mu
pdf("Plots/mu_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(mu[good,1]),xlim=c(-3,3),ylim=c(0,1.3),main=expression(paste
  ("Posterior Distributions for ",mu[i])),type="n")
for (i in 1:ninst){
  lines(density(mu[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)),lty=1,col=inst.col,lwd=3)
dev.off()

# reorder mu by ll
re.mu<-mu[,order(INST[,3])]

# number of evals per instructor
total.nevals<-rep(NA,ninst)
for(i in 1:ninst){
  total.nevals[i]<-sum(courseBYinst[courseBYinst[,1]==unique(inst)[order(
    INST[,3])][i],3])
}

pdf("Plots/mu_post2.pdf")
par(mfrow=c(1,1),las=1)
plot(density(re.mu[good,1]),xlim=c(-3,3),ylim=c(0,1.35),main=expression(
  paste("Posterior Distributions for ",mu[i])),type="n")
for (i in 1:ninst){
  lines(density(re.mu[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)[order(INST[,3])]),lty=1,col=
  inst.col,lwd=3)
#legend("topleft",as.character(total.nevals),lty=1,col=inst.col,lwd=3)
dev.off()

# check old posterior against new ones
par(mfrow=c(1,1),las=1)

```

```

plot(density(re.mu[good,1]),xlim=c(-3,3),ylim=c(0,1.3),main=expression(
  paste("Posterior Distributions for ",mu[i])),type="n")
for (i in 1:ninst){
  lines(density(re.mu[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)[order(INST[,3])]),lty=1,col=
  inst.col,lwd=3)
re.mu_old<-mu_old[,order(INST[,3])]
for (i in 1:ninst){
  lines(density(re.mu_old[good,i]),col=inst.col[i],lwd=2,lty=2)
}

# posterior distributions for sigma^2
# change bw in density to get negatives out?
pdf("Plots/sig2_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(sig2[good,1]),xlim=c(0,1),ylim=c(0,14),main=expression(paste
  ("Posterior Distributions for ",sigma[i]^2)),type="n")
for (i in 1:ninst){
  lines(density(sig2[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)),lty=1,col=inst.col,lwd=3)
dev.off()

# reorder sig2 by ll
re.sig2<-sig2[,order(INST[,3])]
pdf("Plots/sig2_post2.pdf")
par(mfrow=c(1,1),las=1)
plot(density(re.sig2[good,1]),xlim=c(0,1),ylim=c(0,14),main=expression(
  paste("Posterior Distributions for ",sigma[i]^2)),type="n")
for (i in 1:ninst){
  lines(density(re.sig2[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)[order(INST[,3])]),lty=1,col=
  inst.col,lwd=3)
#legend("topleft",as.character(total.nevals),lty=1,col=inst.col,lwd=3)
dev.off()

##### posterior stats
# beta
beta.mean<-apply(bnew,2,mean)
beta.var<-var(bnew)

# mu and sigma^2
mu.mean<-apply(mu,2,mean)
sig.mean<-apply(sig2,2,mean)
mu.var<-apply(mu,2,var)

```

```

sig.var<-apply(sig2,2,var)

# gamma
apply(gamma,2,mean)
apply(gamma,2,var)

# z
apply(znew,2,mean)

post.stats<-cbind(c(beta.mean,NA,NA,NA),c(diag(beta.var),NA,NA,NA),mu.mean
, mu.var,sig.mean,sig.var)
xtable(post.stats,)

##### Calculate proportions for prior predicitive distribution
#gpa<-seq(1,4,length=100)
#pts<-c(4,3,2,1,0)
#size<-c(10,25,50)

prop.post<-as.data.frame(matrix(NA,ncol=length(gpa),nrow=(length(pts)*
length(size))))
rownames(prop.post)<-c("A.10","B.10","C.10","D.10","E.10","A.25","B.25","C
.25","D.25","E.25","A.50","B.50","C.50","D.50","E.50")
colnames(prop.post)<-round(gpa,2)
for(l in 1:length(size)){
  for(i in 1:length(pts)){
    for(j in 1:length(gpa)){
      gam<-rnorm(nn,mu[,1],sig2[,1])
      p<-pnorm(gam + c(size[l],pts[i],gpa[j],pts[i]*gpa[j]))%*%t(bnew[
good,])
      draws<-rbinom(nn,1,p)
      prop.post[i+(5*(l-1)),j]<-table(draws)[2]/nn
    }
  }
}

pdf("Plots/post_pred10.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
="Proportion of Students Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
dev.off()

pdf("Plots/post_pred10_title.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
="Proportion of Students Rating 1", xlab="GPA",main="Class Size 10",cex
.main=2)
color<-rep(purple,3)

```



```

ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)
dev.off()

pdf("Plots/post_pred25.pdf")
par(mfrow=c(1,1),las=1)
# class size 25
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA",main="Class Size 25",cex
     .main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)
dev.off()

pdf("Plots/post_pred50.pdf")
par(mfrow=c(1,1),las=1)
# class size 50
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA",main="Class Size 50",cex
     .main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 11:15){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)
dev.off()

pdf("Plots/post_pred10_50.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=1,col=color[i],lwd=3)
}
for(i in 11:15){
  lines(gpa,prop.post[i,],lty=2,col=color[i],lwd=3)
}
dev.off()

pdf("Plots/post_pred25_2.pdf")
par(mfrow=c(1,1),las=1)
# class size 25

```

```

plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
dev.off()

pdf("Plots/post_pred10_25.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=2,col="gray87",lwd=3)
}
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=1,col=color[i],lwd=3)
}
dev.off()

pdf("Plots/post_pred10_25_50.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA")
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=2,col="gray87",lwd=3)
}
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=1,col="gray87",lwd=3)
}
for(i in 11:15){
  lines(gpa,prop.post[i,],lty=4,col=color[i],lwd=3)
}
dev.off()

pdf("Plots/post_pred3.pdf",height=10,width=3)
par(mfrow=c(3,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA",main="Class Size 10",cex
     .main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)

```

```

# class size 25
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA",main="Class Size 25",cex
     .main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)

# class size 50
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),ylab
     ="Proportion of Students Rating 1", xlab="GPA",main="Class Size 50",cex
     .main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 11:15){
  lines(gpa,prop.post[i,],lty=ltype[i],col=color[i],lwd=3)
}
legend("topright",c("A","B","C","D","E"),lty=1,col=color[1:5],lwd=2)
dev.off()

pdf("Plots/post_pred10_25_50_Title.pdf")
par(mfrow=c(1,1),las=1)
# class size 10
plot(gpa,prop.post[1,],type="n",ylim=c(min(prop.post),max(prop.post)),
     xlab="GPA",ylab="",main="Posterior Predictive",cex.main=2)
color<-rep(purple,3)
ltype<-c(rep(2,5),rep(1,5),rep(4,5))
for(i in 1:5){
  lines(gpa,prop.post[i,],lty=2,col="gray87",lwd=3)
}
for(i in 6:10){
  lines(gpa,prop.post[i,],lty=1,col="gray87",lwd=3)
}
for(i in 11:15){
  lines(gpa,prop.post[i,],lty=4,col=color[i],lwd=3)
}
dev.off()

# intervals for betas
b1<-quantile(bnew[good,1],c(.025,.975))
b2<-quantile(bnew[good,2],c(.025,.975))
b3<-quantile(bnew[good,3],c(.025,.975))
b4<-quantile(bnew[good,4],c(.025,.975))

xtable(cbind(beta.mean,rbind(b1,b2,b3,b4)),digits=4)

# intervals for difference in mu_i's
diff<-matrix(NA,nrow=nn,ncol=21)
CI<-as.data.frame(matrix(NA,nrow=21,ncol=3))

```

```

colnames(CI)<-c("2.5%","97.5%","Significant")
ind<-1
for(i in 1:(ncol(mu)-1)){
  for(j in (i+1):ncol(mu)){
    diff[,ind]<-mu[good,i]-mu[good,j]
    CI[ind,1:2]<-quantile(diff[,ind],c(.025,.975))
    CI[ind,3]<-ifelse((CI[ind,1]*CI[ind,2])>0,1,0)
    rownames(CI)[ind]<-paste(unique(inst)[i],"-",unique(inst)[j])
    ind<-ind+1
  }
}
CI

```

```

beta.mean<-apply(bnew,2,mean)
pnorm(beta.mean)

# intervals for betas
b1<-quantile(bnew[good,1],c(.025,.975))
b2<-quantile(bnew[good,2],c(.025,.975))
b3<-quantile(bnew[good,3],c(.025,.975))
b4<-quantile(bnew[good,4],c(.025,.975))

xtable(cbind(beta.mean,pnorm(beta.mean),rbind(b1,b2,b3,b4)),digits=4)

load("/Users/allisonmbutler/Documents/Classes/Project/Data_651/evalsAB.
  Rdata")
attach(evals)

ss<-table(ScoOverall_Instructo)/nrow(evals)
xtable(t(ss*100),digits=2)

```

C.2 CODE FOR ORDINAL HIERARCHICAL PROBIT MODEL

```

setwd("~/Documents/Classes/Project")
evals2<-read.table("~/Documents/Classes/Project/Data_651/evals2.txt",
  header=T)

# sort evals2 by department, instructor, course
sevals<-NULL
imean_score<-sort(tapply(evals2$ScoOverall_Instructo,
  evals2$bRated_Instructor,mean))
uinst<-as.numeric(names(imean_score))
for(i in 1:length(uinst)){
  iii<-evals2[evals2$bRated_Instructor==ii[i],]
  jj<-sort(unique(iii$bcourse_name))
  for(j in 1:length(jj)){
    jjj<-iii[iii$bcourse_name==jj[j],]
    sevals<-rbind(sevals,jjj)
  }
}

```

```

#sevals[,c(14,17,16)]
write.table(sevals,"Data_651/sevalsCollege9Dept5.txt",row.names=F)

load("/Users/allisonmbutler/Documents/Classes/Project/ordinal_noDept5_10.
Rdata")
setwd("~/Documents/Classes/Project")
library(xtable)

# Function to calculate standard normal distribution
phi<-function(x){
  out<-x
  for(i in 1:length(x)){
    out[i]<-integrate(dnorm,-Inf,x[i])$value
  }
  return(out)
}

evals2<-read.table("~/Documents/Classes/Project/Data_651/
sevalsCollege9Dept5.txt",header=T)
attach(evals2)
n<-nrow(evals2)

# rename instructor variable
inst<-bRated_Instructor
uinst<-unique(inst)
# number of instructors
ninst<-length(unique(inst))
# calculate number of courses each instructor taught and average score
ncourse<-inst.mean<-rep(NA,ninst)
for (i in 1:ninst){
  temp<-evals2[inst==unique(inst)[i],]
  ncourse[i]<-length(unique(temp$courseID))
  inst.mean[i]<-mean(ScoOverall_Instructo[inst==unique(inst)[i]])
}
INST<-cbind(unique(inst), ncourse, inst.mean)

grades<-c("A","A-","B+","B","B-","C+","C","C-","D+","D","D-","E")
points<-c(4,3.7,3.4,3,2.7,2.4,2,1.7,1.4,1,.7,0)
points2<-sort(c(0,2:12),decreasing=T)/3

# change letter grades to points
grade.mat<-matrix(NA,ncol=length(grades),nrow=n)
for(i in 1:length(grades)){
  grade.mat[,i]<-ifelse(Student_Grade==grades[i],points2[i],100)
}
Student_Points<-apply(grade.mat,1,min)
# Note: there are no D grades or 1 pt

# create unique course variable (course and instructor)
course.instID<-paste(bcourse_name,inst,sep=".")
temp<-unlist(strsplit(unique(course.instID),"\\"))
course2<-temp[seq(1,sum(ncourse)*2,by=2)]
inst2<-temp[seq(2,sum(ncourse)*2,by=2)]
# summary stats for course by instructor

```

```

courseBYinst<-as.data.frame(matrix(NA,nrow=sum(ncourse),ncol=12))
courseBYinst[,1]<-inst2
courseBYinst[,2]<-course2
names(courseBYinst)<-c("instructor","course","nevals","credit_hours","
  grade","age","genderM","ClassStanding","GPA","score","InstStatus","
  married")
nevals<-rep(NA,sum(ncourse))
for(i in 1:sum(ncourse)){
  temp<-evals2[course.instID==unique(course.instID)[i],]
  courseBYinst[i,3]<-nevals[i]<-nrow(temp)
  courseBYinst[i,4]<-mean(temp$credit_hours)
  courseBYinst[i,5]<-mean(Student_Points[course.instID==unique(course.
    instID)[i]])
  courseBYinst[i,6]<-mean(temp$Age)
  courseBYinst[i,7]<-mean(temp$Gender=="Male")
  courseBYinst[i,8]<-mean(temp$Class_Standing)
  courseBYinst[i,9]<-mean(temp$Total_GPA)
  courseBYinst[i,10]<-mean(temp$ScoOverall_Instructo)
  courseBYinst[i,11]<-mean(temp$bInstructor_status)
  courseBYinst[i,12]<-mean(temp$Marital_Status=="M")
}
courseBYinst

### Covariates
x<-cbind(Total_Enrolled,Student_Points,Total_GPA, Student_Points*Total_GPA
)
colnames(x)[ncol(x)]<-"Grade X GPA"

# center continuous covariates
mSize<-mean(Total_Enrolled)
mPoints<-mean(Student_Points)
mGPA<-mean(Total_GPA)

xcent<-cbind(Total_Enrolled-mean(Total_Enrolled),Student_Points-mean(
  Student_Points),Total_GPA-mean(Total_GPA), (Student_Points-mean(
  Student_Points))*(Total_GPA-mean(Total_GPA)))
colnames(xcent)<-c("Total_Enrolled","Student_Points","Total_GPA","Grade X
  GPA")

# create instructor score
y1<-ifelse(ScoOverall_Instructo<5,1,ScoOverall_Instructo)
y2<-ifelse(y1==5 | y1==6,2,y1)
y3<-ifelse(y2==7,3,y2)
y<-ifelse(y3==8,4,y3)
#table(ScoOverall_Instructo,y)

y<-ScoOverall_Instructo

#####
library(MASS)
library(MCMCpack)
##### Prior Predictive
set.seed(645)
nn<-10000

```

```

# mu_hi N(m,s^2)
m<-0
s2<-3
xx<-seq(-5,5,length=100)
#plot(xx,dnorm(xx,m,sqrt(s2)),type="l")
mu.pred<-rnorm(nn,m,sqrt(s2))
pnorm(0,m,sqrt(s2))

# sigma^2_hi IG(a,b)
a<-10
b<-5
xx<-seq(0,5,length=100)
#plot(xx,dinvgamma(xx,a,scale=b),type="l")
sig2.pred<-rinvgamma(nn,a,scale=b)
#mean
b/(a-1)
# variance
(b^2)/((a-1)^2*(a-2))
mean(sig2.pred<1)

# theta_hij N(mu_hi, sigma2_hi)
gamma.pred<-rnorm(nn,mu.pred,sqrt(sig2.pred))
#plot(density(gamma.pred))

# beta N(d,g)
nbeta<-ncol(xcent)
d<-rep(0,nbeta)
g<-diag(4,nbeta)
ginv<-solve(g)
beta.pred<-mvrnorm(nn,mu=d, Sigma=g)
pnorm(d)

beta.tab<-matrix(NA,nrow=nbeta,ncol=2)
colnames(beta.tab)<-c("mean","prob < 1")
#par(mfrow=c(3,3),las=1)
for(i in 1:nbeta){
  #plot(density(beta.pred[,i]),main=paste(expression(beta_),i))
  beta.tab[i,1]<-mean(beta.pred[,i])
  beta.tab[i,2]<-mean(beta.pred[,i]<1)
}
beta.tab
var((beta.pred))

# theta Unif
theta1<-0
theta_draw<-NULL
for(i in 1:nn){
  theta_draw<-rbind(theta_draw,sort(runif(cc-2,theta1,4)))
}
theta.pred<-cbind(-Inf,theta1,theta_draw,Inf)

##### Calculate proportions for prior predicitive distribution
# class size: small, medium, and large

```

```

size<-c(10,25,50)-mSize
# grade points
pts<-c(4,3,2,1,0)-mPoints
# total GPA
gpa<-c(1,2,3,4)-mGPA
#gpa<-seq(1,4,length=10)-mGPA
# interaction, grade x gpa

nnn<-nrow(beta.pred)
prop<-as.data.frame(matrix(NA,ncol=(length(gpa)*length(pts)*length(size)),
  nrow=(cc+3)))
index_prior<-1
for(l in 1:length(size)){
  for(i in 1:length(pts)){
    for(j in 1:length(gpa)){
      p<-matrix(NA,nrow=nn,ncol=(cc))
      for(ii in 1:cc){
        p[,ii]<-t(pnorm(theta.pred[, (ii+1)] - gamma.pred - c(size[l],
          pts[i],gpa[j],pts[i]*gpa[j]))**t(beta.pred))) - t(pnorm(
          theta.pred[,ii] - gamma.pred - c(size[l],pts[i],gpa[j],pts[
            i]*gpa[j]))**t(beta.pred)))
      }
      draws<-rep(0,cc)
      for(jj in 1:nnn){
        draws<-draws+rmultinom(1,1,p[jj,])
      }
      prop[-c(1:3),index_prior]<-draws/nnn
      prop[c(1:3),index_prior]<-c(size[l]+mSize,pts[i]+mPoints,gpa[j]+
        mGPA)
      colnames(prop)[index_prior]<-paste(size[l]+mSize,pts[i]+mPoints,
        gpa[j]+mGPA,sep=".")
      index_prior<-index_prior+1
    }
  }
}

```

```

purple1<-rgb(53,0,106,max=255)
purple2<-rgb(122,1,119,max=255)
purple3<-rgb(174,1,126,max=255)
purple4<-rgb(221,52,151,max=255)
purple5<-rgb(247,104,161,max=255)
purple<-c(purple1,purple2,purple3,purple4,purple5)

```

```

par(mfrow=c(4,3),las=1)
for(i in 1:3){
  for(j in 1:4){
    temp<-t(prop[-c(1:3),prop[1,]==(size[i]+mSize) & prop[3,]==(gpa[j]+
      mGPA)])
    barplot(temp, main=paste("Class Size:", size[i]+mSize," \n GPA: ",
      gpa[j]+mGPA),xlab=" Score", col=purple,beside=TRUE,names.arg=(1:8),
      ylim=c(0,(max(temp)+.05)))
    box()
    #legend("topleft",c("A","B","C","D","E"), fill=purple)
  }
}

```



```

}
}

pdf("PlotsOrdNoDept/prior_pred10.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop[-c(1:3),prop[1,]==10 & prop[3,]==3])
barplot(temp, main="Class Size: 10 \n GPA: 3",xlab="Score", col=purple,
  beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("center",c("A","B","C","D","E"), fill=purple)
dev.off()

pdf("PlotsOrdNoDept/prior_pred25.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop[-c(1:3),prop[1,]==25 & prop[3,]==3])
barplot(temp, main="Class Size: 25 \n GPA: 3",xlab="Score", col=purple,
  beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("center",c("A","B","C","D","E"), fill=purple)
dev.off()

pdf("PlotsOrdNoDept/prior_pred50.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop[-c(1:3),prop[1,]==50 & prop[3,]==3])
barplot(temp, main="Class Size: 50 \n GPA: 3",xlab="Score", col=purple,
  beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("center",c("A","B","C","D","E"), fill=purple)
dev.off()

##### instructor by mean scores
courseBYinst[,c(1:3,10)]
ordered.inst<-INST[order(INST[,3]),]

# proportion of 1's for ordered instructors
prop1<-rep(NA,ninst)
for(i in 1:ninst){
  prop1[i]<-mean(y[inst==ordered.inst[i,1]])
}

l1<-as.data.frame(matrix(NA,nrow=ninst,ncol=(max(ordered.inst[,2])+4)))
names(l1)<-c("Instructor","ncourse","inst.mean","prop1","course1","course2",
  "course3","course4","course5","course6")
sdd<-rep(NA,ninst)
for(i in 1:ninst){
  l1<-courseBYinst$score[courseBYinst[,1]==ordered.inst[i,1]]
  l1[i,(1:(4+length(l1)))]<-c(ordered.inst[i,1:3],prop1[i],l1)
  sdd[i]<-sd(l1)
}

xtable(l1[,-2])

xtable(cbind(l1[,c(1,3,4)],sdd))

```

```

#####
library(msm)
set.seed(22)
burn<-10000
nn<-100000
nsim<-(nn+burn)

#number of categories
cc<-length(unique(y))

#comment<-FALSE
#if(comment==F){
bnew<-matrix(d,nrow=nsim+1,ncol=nbeta,byrow=T)
znew<-matrix(0,nrow=nsim+1,ncol=n,byrow=T)
mu<-matrix(rnorm(ninst,m,sqrt(s2)),nrow=nsim+1,ncol=ninst,byrow=T)
sig2<-matrix(rinvgamma(ninst,a,b),nrow=nsim+1,ncol=ninst,byrow=T)
gam_init<-NULL
for (i in 1:ninst){
  gam_init<-c(gam_init,rnorm(ncourse[i],mu[1,i],sqrt(sig2[1,i])))
}
gamma<-matrix(gam_init,nrow=nsim+1,ncol=sum(ncourse),byrow=T)
theta_init<-seq(-1,cc-1,by=1)
theta_init[1]<--Inf
theta_init[cc+1]<-Inf
theta<-matrix(theta_init,nrow=nsim+1,ncol=cc+1,byrow=T)
sigMH<-0.04
index<-c(0:cc)
ssinv<-solve(ginv + t(xcent)%*%xcent)
ar<-0

for (l in 2:(nsim+1)){
#l<-2
  # enlongate gamma
  gam2<-NULL
  for(i in 1:sum(ncourse)){
    gam2<-c(gam2,rep(gamma[l-1,i],nevals[i]))
  }

  ##### update theta
  gg<-theta[1,]
  for (j in 2:(cc-1)){
    g<-rtnorm(1,theta[l-1,index==j],sigMH,lower=gg[index==j-1],upper=
      theta[l-1,index==j+1])
    gg[index==j]<-g
  }
  gg[cc+1]<-gg[cc]+4
  linPred<-gam2 + xcent%*%bnew[l-1,]

  temp<-((pnorm(gg[y+1]-linPred) - pnorm(gg[(y)]-linPred)) / (pnorm(theta[l
    -1,y+1]-linPred) - pnorm(theta[l-1,y]-linPred)))

  r1<-prod(temp)
}

```

```

if(sum(temp=="NaN")>0){
  indR<-(1:N)[temp=="NaN"]

  r11<-prod(temp[-indR])

  r12<-prod((phi(gg[y[indR]+1]-linPred[indR]) - phi(gg[(y[indR]])-
    linPred[indR])))/(phi(theta[l-1,y[indR]+1]-linPred[indR]) - phi(
    theta[l-1,y[indR]]-linPred[indR])))

  r1<-r11*r12
}

r2<-prod((pnorm((theta[l-1,(3+1):(cc+1)]-theta[l-1,(2+1):(cc)])/sigMH)-
  pnorm((gg[2:(cc-1)]-theta[l-1,3:cc])/sigMH))/(pnorm((gg[4:(cc+1)]-gg
  [3:cc])/sigMH)-pnorm((theta[l-1,2:(cc-1)]-gg[3:cc])/sigMH)))

r<-r1*r2
u<-runif(1,0,1)
if (u<r){
  theta[l,]<-gg
  ar<-ar+1
} else{
  theta[l,]<-theta[l-1,]
}

##### update beta
bnew[l,]<-mvrnorm(1,mu= ssinv%*(ginv%*d + t(xcent)%*(znew[l-1,]-gam2
  )),Sigma=ssinv)

##### update z
low<-theta[l,y]
up<-theta[l,y+1]
znew[l,]<-rtnorm(n,mean=(gam2+xcent%*bnew[l,]),sd=1,lower=low,upper=up
  )

##### update mu
# sum gamma by instructor
gamSum<-NULL
for(i in 1:ninst){
  gamSum<-c(gamSum,sum(gamma[l-1,as.numeric(courseBYinst[,1])]==unique(
    inst)[i])))
}

denom<-sig2[l,] + s2*ncourse
num1<-sig2[l,]*m + s2*gamSum
num2<-sig2[l,]*s2
mu[l,]<-rnorm(ninst,num1/denom,sqrt(num2/denom))
##### update sig2
# enlongate mu and sig2
mu2<-sig22<-NULL
for(i in 1:ninst){
  mu2<-c(mu2,rep(mu[l,i],ncourse[i]))
  sig22<-c(sig22,rep(sig2[l,i],ncourse[i]))
}

```

```

# calculate sum of squares
squares<-(gamma[l-1,]-mu2)^2
ssgam<-NULL
for(i in 1:ninst){
  ssgam<-c(ssgam,sum(squares[as.numeric(courseBYinst[,1])==uinst[i]]))
}

siga<-a+ncourse/2
sigb<-1/((1/b)+ssgam*.5)
sig2[l,]<-1/rgamma(ninst,siga,scale=sigb)

##### update gamma

# sum of squares
ssz<-NULL
sum1<-znew[l,]-xcent%%bnew[l,]
ind<-cbind(c(1,cumsum(nevals)[-sum(ncourse)]+1),cumsum(nevals))
for(i in 1:sum(ncourse)){
  ssz<-c(ssz,sum(sum1[ind[i,1]:ind[i,2]]))
}

num1<-mu2 + sig22*ssz
denom<-1 + sig22*nevals
gamma[l,]<-rnorm(sum(ncourse),num1/denom,sqrt(sig22/denom))

if(1%%100==0){ print(l) }
}
#}
# To lower ar, increase sigMH
ar/(nsim+1)
good<-(-c(1:(burn+1)))

##### trace plots
small<-c(((nsim+1)-1000):(nsim+1))
# theta
pdf("PlotsOrdNoDept/trace_theta.pdf")
par(mfrow=c(3,2),oma = c(0, 0, 3, 0),las=1)
for(i in 1:6){
  plot(2:(nsim+1),theta[-1,i+2],type="l",xlab="", main =paste("theta[",i
+1,"]",sep=""),cex.main=2,ylab="")
  lines(-good,theta[-good,i+2],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",theta)),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_thetaSmall.pdf")
par(mfrow=c(3,2),oma = c(0, 0, 3, 0),las=1)
for(i in 1:6){
  plot(small,theta[small,i+2],type="l",xlab="", main =paste("theta[",i
+1,"]",sep=""),cex.main=2,ylab="")
}
mtext(expression(paste("Trace Plots for ",theta)),cex=2,outer=T)
dev.off()

```

```

# beta
pdf("PlotsOrdNoDept/trace_beta.pdf",height=5,width=10)
par(mfrow=c(2,2),oma = c(0, 0, 3, 0),las=1)
for(i in 1:nbeta){
  plot(2:(nsim+1),bnew[-1,i],type="l",xlab="", main=colnames(xcent)[i],
    cex.main=2,ylab="")
  lines(-good,bnew[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",beta)),cex=2,outer=T)
dev.off()

mean(bnew[,1]>0)

pdf("PlotsOrdNoDept/trace_betaSmall.pdf")
par(mfrow=c(2,2),oma = c(0, 0, 3, 0),las=1)
for(i in 1:nbeta){
  plot(small,bnew[small,i],type="l",xlab="", main=colnames(xcent)[i],cex.
    main=2,ylab="")
}
mtext(expression(paste("Trace Plots for ",beta)),cex=2,outer=T)
dev.off()

# mu_i
pdf("PlotsOrdNoDept/trace_mu.pdf",height=5,width=10)
par(mfrow=c(2,4),oma = c(0, 0, 3, 0),las=1)
for(i in 1:ninst){
  plot(2:(nsim+1),mu[-1,i],type="l",main=paste("Instructor ",unique(inst)
    [i]),ylab=expression(mu),xlab="",cex.main=2)
  lines(-good,mu[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",mu[i])),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_muDOC.pdf",height=10,width=8)
par(mfrow=c(4,2),las=1,oma = c(0, 0, 3, 0),mar=c(2,3.85,4,1))
for(i in 1:ninst){
  plot(small,mu[small,i],type="l",main=paste("Instructor ",unique(inst)[i]
    ),ylab=expression(mu),xlab="",cex.main=2)
}
mtext(expression(paste("Trace Plots for ",mu[i])),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_muSlides.pdf",height=4,width=10)
par(mfrow=c(2,4),las=1,oma = c(0, 0, 3, 0),mar=c(2,3.85,4,1))
for(i in 1:ninst){
  plot(small,mu[small,i],type="l",main=paste("Instructor ",unique(inst)[i]
    ),ylab=expression(mu),xlab="",cex.main=2)
}
mtext(expression(paste("Trace Plots for ",mu[i])),cex=2,outer=T)
dev.off()

# sigma^2_i

```

```

# Don't look good
pdf("PlotsOrdNoDept/trace_sig2.pdf",width=10,height=5)
par(mfrow=c(2,4),oma = c(0, 0, 3, 0),las=1)
for(i in 1:ninst){
  plot(2:(nsim+1),sig2[-1,i],type="l",main=paste("Instructor ",unique(
    inst)[i]),ylab=expression(paste(sigma^2)),xlab="",cex.main=2)
  lines(-good,sig2[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",sigma[i]^2)),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_sig2DOC.pdf",height=10,width=8)
par(mfrow=c(4,2),las=1,oma = c(0, 0, 4, 0),mar=c(2,3.85,4,1))
for(i in 1:ninst){
  plot(small,sig2[small,i],type="l",main=paste("Instructor ",unique(inst)
    [i]),ylab=expression(paste(sigma^2)),xlab="",cex.main=2)
}
mtext(expression(paste("Trace Plots for ",sigma[i]^2)),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_sig2Slides.pdf",height=4,width=10)
par(mfrow=c(2,4),las=1,oma = c(0, 0, 4, 0),mar=c(2,3.85,4,1))
for(i in 1:ninst){
  plot(small,sig2[small,i],type="l",main=paste("Instructor ",unique(inst)
    [i]),ylab=expression(paste(sigma^2)),xlab="",cex.main=2)
}
mtext(expression(paste("Trace Plots for ",sigma[i]^2)),cex=2,outer=T)
dev.off()

# gamma
pdf("PlotsOrdNoDept/trace_gamma.pdf")
par(mfrow=c(3,3),oma = c(0, 0, 3, 0),las=1)
for(i in 1:9){
  plot(2:(nsim+1),gamma[-1,i],type="l",xlab="",ylab=expression(gamma))
  lines(-good,gamma[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",gamma[ij])),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_gammaDOC.pdf",height=10,width=8)
par(mfrow=c(4,2),las=1,oma = c(0, 0, 4, 0),mar=c(2,3.85,4,1))
for(i in 1:8){
  plot(small,gamma[small,i],type="l",xlab="",ylab=expression(gamma))
}
mtext(expression(paste("Trace Plots for ",gamma[ij])),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_gammaSlides.pdf",height=4,width=10)
par(mfrow=c(2,4),las=1,oma = c(0, 0, 4, 0),mar=c(2,3.85,4,1))
for(i in 1:8){
  plot(small,gamma[small,i],type="l",xlab="",ylab=expression(gamma))
}
mtext(expression(paste("Trace Plots for ",gamma[ij])),cex=2,outer=T)
dev.off()

```

```

# z
pdf("PlotsOrdNoDept/trace_z.pdf")
par(mfrow=c(3,3),oma = c(0, 0, 3, 0),las=1)
for(i in 1:9){
  plot(2:(nsim+1),znew[-1,i],type="l",xlab="",ylab=expression(z))
  lines(-good,znew[-good,i],type="l",col="red")
}
mtext(expression(paste("Trace Plots for ",z[ijk])),cex=2,outer=T)
dev.off()

pdf("PlotsOrdNoDept/trace_zDOC.pdf",height=10,width=8)
par(mfrow=c(4,2),las=1,oma = c(0, 0, 4, 0),mar=c(2,3.85,4,1))
for(i in 1:8){
  plot(small,znew[small,i],type="l",xlab="",ylab=expression(z))
}
mtext(expression(paste("Trace Plots for ",z[ijk])),cex=2,outer=T)
dev.off()

##### Posterior Distributions
# Function to calculate the 95% CI for difference in distributions
CIfunc<-function(data,name){
  # data, matrix with each distribution in columns
  # name, vector of the names of each column
  N<-nrow(data)
  ccc<-0
  for(i in 1:(ncol(data)-1)){
    ccc<-ccc+ncol(data)-i
  }
  diff<-matrix(NA,nrow=N,ncol=ccc)
  CI<-as.data.frame(matrix(NA,nrow=ccc,ncol=3))
  colnames(CI)<-c("2.5%","97.5%","Significant")
  ind<-1
  for(i in 1:(ncol(data)-1)){
    for(j in (i+1):ncol(data)){
      diff[,ind]<-data[,i]-data[,j]
      CI[ind,1:2]<-quantile(diff[,ind],c(.025,.975))
      CI[ind,3]<-ifelse((CI[ind,1]*CI[ind,2])>0,1,NA)
      rownames(CI)[ind]<-paste(name[i],"-",name[j])
      ind<-ind+1
    }
  }
  return(CI)
}

# colors for 7 instructors
inst1<-rgb(27,158,119,max=255)
inst2<-rgb(217,95,2,max=255)
inst3<-rgb(117,112,179,max=255)
inst4<-rgb(231,41,138,max=255)
inst5<-rgb(102,102,102,max=255)
inst6<-rgb(230,171,2,max=255)
inst7<-rgb(128,177,211,max=255)
inst.col<-c(inst1,inst2,inst3,inst4,inst5,inst6,inst7)

```

```
#####
#### posterior distributions for mu
pdf("PlotsOrdNoDept/mu_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(mu[good,1]),xlim=c(-1,5),ylim=c(0,1.6),main=expression(paste
  ("Posterior Distributions for ",mu[i])),type="n",xlab="")
for (i in 1:ninst){
  lines(density(mu[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)),lty=1,col=inst.col,lwd=3)
dev.off()

# number of evals per instructor
total.nevals<-rep(NA,ninst)
for(i in 1:ninst){
  total.nevals[i]<-sum(courseBYinst[courseBYinst[,1]==unique(inst)[order(
    INST[,3])][i],3])
}

# intervals for difference in mu_i's
# reorder mu to go from highest to lowest
mu2<-mu[,c(7,6,5,4,3,2,1)]
uinst2<-uinst[c(7,6,5,4,3,2,1)]

CI<-CIfunc(data=mu2[good,],name=uinst2)
muCI<-cbind(CI[1:11,],c(rownames(CI)[12:21],NA),rbind(CI[12:21,],rep(NA,3)
))
xtable(muCI)

#####
# posterior distributions for sigma^2
pdf("PlotsOrdNoDept/sig2_post.pdf")
par(mfrow=c(1,1),las=1)
plot(density(sig2[good,1]),xlim=c(0,.15),ylim=c(0,29),main=expression(
  paste("Posterior Distributions for ",sigma[i]^2)),type="n",xlab="")
for (i in 1:ninst){
  lines(density(sig2[good,i]),col=inst.col[i],lwd=3)
}
legend("topright",as.character(unique(inst)),lty=1,col=inst.col,lwd=3)
dev.off()

#####
# 7 colors for courses
c1<-rgb(228,26,28,max=255)
c2<-rgb(55,126,184,max=255)
c3<-rgb(77,175,74,max=255)
c4<-rgb(102,102,102,max=255)
c5<-rgb(255,127,0,max=255)
c6<-rgb(231,41,138,max=255)
c7<-rgb(74,20,134,max=255)
c8<-"black"
c.col<-c(c1,c2,c3,c4,c5,c6,c7)
c.col1<-c(c1,c8,c3,c7,c2)
```



```

c.col2<-c(c2,c5,c6,c4)

# gamma for each instructor
pdf("PlotsOrdNoDept/gammaInst_post.pdf",height=5,width=10)
par(mfrow=c(2,4),las=1)
for(j in 1:ninst){
  temp<-gamma[good,courseBYinst[,1]==unique(inst)[j]]
  #par(mfrow=c(1,1))
  maxY<-2.3
  plot(density(temp[,1]),xlim=c(-1,5),ylim=c(0,maxY),main=paste("
  Instructor ",unique(inst)[j]),type="n")
  text(0.1,maxY,"Course:")
  for (i in 1:ncol(temp)){
    lines(density(temp[,i]),col=c.col[i])
    text(.1,(maxY-(.2*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
    ],2][i],col=c.col[i])
  }
}
dev.off()

# gamma for instructor 996
pdf("PlotsOrdNoDept/gammaInst_postBest.pdf")
par(mfrow=c(1,1),las=1)
ii<-996
temp<-gamma[good,courseBYinst[,1]==ii]
plot(density(temp[,1]),xlim=c(1.8,5),ylim=c(0,2.3),main=paste("
  Instructor ",ii),cex.main=2)
#text(-1.3,2,"Course:",cex=1.5)
for (i in 1:ncol(temp)){
  lines(density(temp[,i]),col=c.col1[i],lwd=2)
  #text(-1.3,(2-(.2*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
  ],2][i],col=c.col[i],cex=1.5)
}
legend("topleft",paste("Course",courseBYinst[courseBYinst[,1]==ii,2]),col=
  c.col1,lty=1,lwd=2)
dev.off()

# gamma for instructor 7
pdf("PlotsOrdNoDept/gammaInst_postWorst.pdf")
par(mfrow=c(1,1),las=1)
temp<-gamma[good,courseBYinst[,1]==unique(inst)[1]]
#par(mfrow=c(1,1))
plot(density(temp[,1]),xlim=c(-.6,3.4),ylim=c(0,1.7),main=paste("
  Instructor ",unique(inst)[1]),cex.main=2)
#text(-2,1.5,"Course:",cex=1.5)
for (i in 1:ncol(temp)){
  lines(density(temp[,i]),col=c.col2[i],lwd=2)
  #text(-2,(1.5-(.15*i)),courseBYinst[courseBYinst[,1]==unique(inst)[j
  ],2][i],col=c.col[i],cex=1.5)
}
legend("topleft",paste("Course",courseBYinst[courseBYinst[,1]==unique(inst)
][1],2]),col=c.col2,lty=1,lwd=2)
dev.off()

```

```

# gamma for each course
# calculate number of instructors teaching each course
ninstBYcourse<-unique(bcourse_name)
for(j in 1:length(unique(bcourse_name))){
  ninstBYcourse[j]<-sum(courseBYinst[,2]==unique(bcourse_name)[j])
}
sum(ninstBYcourse>1)
multInstCourses<-unique(bcourse_name)[ninstBYcourse>1]
# change order to match binary model
multInstCourses<-c(4324, 479, 5187, 3527, 4591, 634, 2070 , 1020)

pdf("PlotsOrdNoDept/gammaCourse_post.pdf",height=5,width=10)
par(mfrow=c(2,4),las=1)
for(j in 1:sum(ninstBYcourse>1)){
  temp<-gamma[good,courseBYinst[,2]==multInstCourses[j]]
  #par(mfrow=c(1,1))
  maxY<-2.4
  plot(density(temp[,1]),xlim=c(-1,5),ylim=c(0,maxY),main=paste("Course
", multInstCourses[j]),type="n",cex.main=2)
  text(.1,maxY,"Instructor:",cex=1.5)
  temp.inst<-courseBYinst[courseBYinst[,2]==multInstCourses[j],1]
  for (i in 1:ncol(temp)){
    lines(density(temp[,i]),col=inst.col[unique(inst)[order(INST[,3])]==
temp.inst[i]])
    text(.1,(maxY-(.4*i)),temp.inst[i],col=inst.col[unique(inst)[order(
INST[,3])]==temp.inst[i]],cex=1.5)
  }
}
dev.off()

pdf("PlotsOrdNoDept/gammaCourse_post2.pdf",height=10,width=6)
par(mfrow=c(4,2),las=1,mar=c(2,3.85,4,1))
for(j in 1:sum(ninstBYcourse>1)){
  temp<-gamma[good,courseBYinst[,2]==multInstCourses[j]]
  #par(mfrow=c(1,1))
  maxY<-2.4
  plot(density(temp[,1]),xlim=c(-1,5),ylim=c(0, maxY),main=paste("Course
", multInstCourses[j]),type="n",cex.main=2)
  text(0,maxY,"Instructor:",cex=1.5)
  temp.inst<-courseBYinst[courseBYinst[,2]==multInstCourses[j],1]
  for (i in 1:ncol(temp)){
    lines(density(temp[,i]),col=inst.col[unique(inst)[order(INST[,3])]==
temp.inst[i]])
    text(.1,(maxY-(.3*i)),temp.inst[i],col=inst.col[unique(inst)[order(
INST[,3])]==temp.inst[i]],cex=1.5)
  }
}
dev.off()

# CI for differences in gamma for instructors who taught the same course
gamCI<-NULL
for(i in 1:8){
  course<-multInstCourses[i]

```

```

    CI_temp<-CIfunc(gamma[good, courseBYinst[,2]==course], courseBYinst[
      courseBYinst[,2]==course,1])
    CI_temp<-cbind(rownames(CI_temp), CI_temp)
    rownames(CI_temp)[1]<-paste("Course ", course, sep="")
    gamCI<-rbind(gamCI, CI_temp)
  }
xtable(gamCI)

#####
# calculate average theta
theta_mean<-cbind(apply(theta[,c(2:8)], 2, mean))
rownames(theta_mean)<-c("theta_1", "theta_2", "theta_3", "theta_4", "theta_5",
  "theta_6", "theta_7")
colnames(theta_mean)<-"Mean"
xtable(theta_mean, digits=2)

par(mfrow=c(3,2), las=1, mar=c(2,3.85,4,1))
for(i in 3:(cc)){
  plot(density(theta[good,i]), xlab="", main=paste("theta[", i, "]", sep=""))
}

#####
# colors for 8 scores
s1<-rgb(16,169,207,max=255)
s1<-rgb(27,205,187,max=255)
s2<-rgb(65,182,196,max=255)
s3<-rgb(29,145,192,max=255)
s4<-rgb(34,94,168,max=255)
s5<-rgb(37,52,148,max=255)
s6<-rgb(5,82,96,max=255)
s7<-rgb(8,29,88,max=255)
s8<-rgb(0,0,20,max=255)
score.col<-c(s1,s2,s3,s4,s5,s6,s7,s8)

plot(1,1,ylim=c(0,cc),xlim=c(0,cc),type="n")
for(i in 1:cc){
  abline(v=i,col=score.col[i])
}

pdf("PlotsOrdNoDept/z_course.pdf")
par(oma = c(0, 0, 3, 0))
x <- rbind(cbind(rbind(1,3),rbind(2,4), 5))
layout(x, width = c(8,8,2))
#layout.show(5)
#par(mfrow=c(2,2),oma = c(0, 0, 3, 0),las=1)
# instructor 3086
tt<-(1:nrow(evals2))[bRated_Instructor==3086 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 3086",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]
  n1[i]<-length(ii)
}

```

```

    if(length(ii)>0){
      for(j in 1:length(ii)){
        lines(density(znew[good,ii[j]]),col=score.col[indd])
      }
    }
    indd<-indd+1
  }
#legend("topright",as.character(c(1:cc)),lty=1,col=score.col,cex=.8)
score1<-evals2[tt,]$ScoOverall_Instructo
n.score1<-rep(NA,cc)
for(i in 1:cc){
  n.score1[i]<-sum(score1==i)
}
text(4.8,2.5,paste("N=",length(tt),sep=""))
for(i in 1:cc){
  text(5,2.5-(i*.175),paste("n[" ,i, " ] = ",n.score1[i],sep=""))
}

abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
[2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)

# instructor 996
tt<-(1:nrow(evals2))[bRated_Instructor==996 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 996",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]
  n1[i]<-length(ii)
  if(length(ii)>0){
    for(j in 1:length(ii)){
      lines(density(znew[good,ii[j]]),col=score.col[indd])
    }
  }
}
indd<-indd+1
}
#legend("topleft",as.character(c(1:cc)),lty=1,col=score.col,cex=.8)
score1<-evals2[tt,]$ScoOverall_Instructo
n.score1<-rep(NA,cc)
for(i in 1:cc){
  n.score1[i]<-sum(score1==i)
}
text(4.8,2.5,paste("N=",length(tt),sep=""))
for(i in 1:cc){
  text(5,2.5-(i*.175),paste("n[" ,i, " ] = ",n.score1[i],sep=""))
}

abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
[2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)

```

```

# instructor 2847
tt<-(1:nrow(evals2))[bRated_Instructor==2847 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 2847",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]
  n1[i]<-length(ii)
  if(length(ii)>0){
    for(j in 1:length(ii)){
      lines(density(znew[good,ii[j]]),col=score.col[indd])
    }
  }
  indd<-indd+1
}
#legend("topleft",as.character(c(1:cc)),lty=1,col=score.col,cex=.8)
score1<-evals2[tt,]$ScoOverall_Instructo
n.score1<-rep(NA,cc)
for(i in 1:cc){
  n.score1[i]<-sum(score1==i)
}
text(4.8,2.5,paste("N=",length(tt),sep=""))
for(i in 1:cc){
  text(5,2.5-(i*.175),paste("n[",i,"] = ",n.score1[i],sep=""))
}

abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
  [2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
  expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)

# instructor 4359
tt<-(1:nrow(evals2))[bRated_Instructor==4359 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 4359",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]
  n1[i]<-length(ii)
  if(length(ii)>0){
    for(j in 1:length(ii)){
      lines(density(znew[good,ii[j]]),col=score.col[indd])
    }
  }
  indd<-indd+1
}
score1<-evals2[tt,]$ScoOverall_Instructo
n.score1<-rep(NA,cc)
for(i in 1:cc){
  n.score1[i]<-sum(score1==i)
}

```

```

text(4.8,2.5,paste("N=",length(tt),sep=""))
for(i in 1:cc){
  text(5,2.5-(i*.175),paste("n[",i,"] = ",n.score1[i],sep=""))
}

abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
  [2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
  expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)

oldMar <- par(mar = c(0,0,0,0))
plot.new()
legend("topleft",as.character(c(1:cc)),lty=1,col=score.col,cex=1.5)
par(oldMar)

mtext("Course 3527",cex=1.5,outer=T)
dev.off()

pdf("PlotsOrdNoDept/z_courseBad.pdf")
par(mfrow=c(1,1),las=1)
# instructor 3086
tt<-(1:nrow(evals2))[bRated_Instructor==3086 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 3086",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]
  n1[i]<-length(ii)
  if(length(ii)>0){
    for(j in 1:length(ii)){
      lines(density(znew[good,ii[j]]),col=score.col[indd])
    }
  }
  indd<-indd+1
}
legend("topright",as.character(c(1:cc)),lty=1,col=score.col,cex=.8)
abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
  [2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
  expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)
dev.off()

pdf("PlotsOrdNoDept/z_courseGood.pdf")
par(mfrow=c(1,1),las=1)
# instructor 996
tt<-(1:nrow(evals2))[bRated_Instructor==996 & bcourse_name==3527]
plot(density(znew[good,tt[1]]),ylim=c(0,2.5),xlim=c(-1.5,5.5),main="
  Instructor 996",xlab="",type="n")
indd<-1
n1<-rep(NA,8)
for(i in 1:(cc)){
  ii<-tt[y[tt]==i]

```

```

n1[i]<-length(ii)
if(length(ii)>0){
  for(j in 1:length(ii)){
    lines(density(znew[good,ii[j]]),col=score.col[indd])
  }
}
indd<-indd+1
}
legend("topright",as.character(c(1:cc)),lty=1,col=score.col,cex=.8)
abline(v=theta_mean,lty=2)
axis(side=3,at=theta_mean,labels=c(expression(theta[1]),expression(theta
[2]),expression(theta[3]),expression(theta[4]),expression(theta[5]),
expression(theta[6]),expression(theta[7])),cex=2,tick=F,line=-.8)
dev.off()

#####
beta.mean<-apply(bnew,2,mean)
pnorm(beta.mean)

# intervals for betas
bCI<-matrix(NA,nrow=nbeta,ncol=2)
for(i in 1:nbeta){
  bCI[i,]<-quantile(bnew[good,i],c(.025,.975))
}
#betaCI<-cbind(beta.mean,pnorm(beta.mean),bCI)
#colnames(betaCI)<-c("mean","p(mean)","2.5%","97.5%")
betaCI<-cbind(beta.mean,bCI)
colnames(betaCI)<-c("mean","2.5%","97.5%")
rownames(betaCI)<-colnames(xcent)
xtable(betaCI,digits=4)

##### Calculate proportions for prior predicitive distribution
#gpa<-seq(1,4,length=100)
#pts<-c(4,3,2,1,0)
#size<-c(10,25,50)
# use instructor 2847
nn<-length(gamma[good,1])

prop.post<-as.data.frame(matrix(NA,ncol=(length(gpa)*length(pts)*length(
size)),nrow=(cc+3)))
rownames(prop.post)<-c("size","points","GPA",1:8)
index_post<-1
gam<-rnorm(nn,mu[good,3],sig2[good,3])
for(l in 1:length(size)){
  for(i in 1:length(pts)){
    for(j in 1:length(gpa)){
      p<-matrix(NA,nrow=nn,ncol=(cc))
      for(ii in 1:cc){
        p[,ii]<-t(pnorm(theta[good,(ii+1)] - gam - c(size[l],pts[i],
gpa[j],pts[i]*gpa[j]))%*%t(bnew[good,]))) - t(pnorm(theta[
good,(ii)] - gam - c(size[l],pts[i],gpa[j],pts[i]*gpa[j])
%*%t(bnew[good,])))
      }
    }
  }
}

```

```

        draws<-rep(0,cc)
        for(jj in 1:nn){
            draws<-draws+rmultinom(1,1,p[jj,])
        }
        prop.post[-c(1:3),index_post]<-draws/nn
        prop.post[c(1:3),index_post]<-c(size[l]+mSize,pts[i]+mPoints,gpa[
            j]+mGPA)
        colnames(prop.post)[index_post]<-paste(size[l]+mSize,pts[i]+
            mPoints,gpa[j]+mGPA,sep=".")
        index_post<-index_post+1
    }
}

par(mfrow=c(4,3),las=1)
for(i in 1:3){
    for(j in 1:4){
        temp<-t(prop.post[-c(1:3),prop.post[1,]==(size[i]+mSize) & prop.post
            [3,]==(gpa[j]+mGPA)])
        barplot(temp, main=paste("Class Size:", size[i]+mSize," \n GPA: ",
            gpa[j]+mGPA),xlab="Score", col=purple,beside=TRUE,names.arg=(1:8)
            ,ylim=c(0,(max(temp)+.05)))
        box()
        #legend("topleft",c("A","B","C","D","E"), fill=purple)
    }
}

pdf("PlotsOrdNoDept/post_pred10.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop.post[-c(1:3),prop.post[1,]==10 & prop.post[3,]==3])
barplot(temp, main="Class Size: 10 \n GPA: 3",xlab="Score", col=purple,
    beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("topleft",c("A","B","C","D","E"), fill=purple)
dev.off()

pdf("PlotsOrdNoDept/post_pred25.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop.post[-c(1:3),prop.post[1,]==25 & prop.post[3,]==3])
barplot(temp, main="Class Size: 25 \n GPA: 3",xlab="Score", col=purple,
    beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("topleft",c("A","B","C","D","E"), fill=purple)
dev.off()

pdf("PlotsOrdNoDept/post_pred50.pdf")
par(mfrow=c(1,1),las=1)
temp<-t(prop.post[-c(1:3),prop.post[1,]==50 & prop.post[3,]==3])
barplot(temp, main="Class Size: 50 \n GPA: 3",xlab="Score", col=purple,
    beside=TRUE,names.arg=(1:8),ylim=c(0,(max(temp)+.05)))
box()
legend("topleft",c("A","B","C","D","E"), fill=purple)
dev.off()

```