A Statistical Model of Student Knowledge for a Corrected Conceptual Gain

An Honors Thesis submitted in partial fulfillment
of the requirements of Honors Studies in Physics

By
Edward Corcoran

Spring 2005

Physics
J. William Fulbright College of Arts and Sciences
University of Arkansas

Acknowledgements

Many thanks to Dr. John Stewart for giving me the chance to participate in the research project and for his constant guidance and boundless patience.

Thanks also to Dr. Gay Stewart for her assistance on the project and for helping me with all the issues that come with being a physics major.

Table of Contents

Abstract

A student's state of knowledge entering a science class is variable. Some students have no prior knowledge of the material, where other students have substantial previous knowledge. The distribution of these states of knowledge is extracted by fitting test data with combinations of the binomial distribution. Since the pretest scores of both the Force Concepts Inventory and the Conceptual Survey of Electricity and Magnetism contain a substantial number of questions that appear to be answered simply by random guessing the multiple choice pretest overestimates the real state of student knowledge leading to a bias in the conceptual gain and the normalized conceptual gain towards low values. This bias can be removed from the conceptual gain by correcting the pretest score for the effect of guessing and constructing an effective present score. This effective pretest is an estimate of what the student would score on a non-multiple choice test.

Chapter 1: INTRODUCTION

Within the greater educational reform movement, one of the most complex problems is how best to teach. Funding can be increased and redistributed, teachers can be retrained, and school districts reorganized, but to turn out better taught kids, the proclaimed goal of all ideas for educational reform, at some point teaching has to improve. One promising idea for improving teaching has been increasing the interactivity in the classroom. In different subjects, this takes different forms, but in the sciences it often involves more hands-on learning and more interaction between students and between the student and the teacher.

In traditionally taught physics classes, the focus has been on memorizing equations and using those to solve problems. While many students are able to perform well at this, they often do not truly understand the physics behind the equations. The way physics works often contradicts with the way they think the world works, and when students are asked to think about new physics situations, they use their intuition and life experiences, not what they have been taught in physics. A possible solution to this is the interactive teaching method, which tries to connect the rules of physics with students real life experiences so that they understand what happens at a deeper level. This fosters a real conceptual understand of physics, instead of just a knowledge of some equations and how to plug numbers into them to get an answer.

The advantages of the interactive teaching method have to be proved so that it will be widely adopted by educators. Initial results are promising and indicate that reform methods produce students who have a better conceptual understanding of physics. Within the physics education community, the two most common tests for measuring the skill level of introductory physics students are the Force Concepts Inventory (FCI) (Hestenes) which covers mechanics and the Conceptual Survey of Electricity and Magnetism (CSEM) (Maloney) which covers electricity and magnetism. These tests are used to gauge how much conceptual understanding a student has gained while taking a class. Each test is given at the beginning and end of the semester, and the student's increase in conceptual understanding is characterized by the Normalized Conceptual Gain or Hake Gain. This allows the comparison of different teaching methods without penalizing schools whose students come in with little or no physics knowledge.

The Hake Gain is defined as the actual gain of student divided by the maximum possible gain, or

$$HakeGain \equiv \frac{(posttest)\% - (pretest)\%}{100\% - (pretest)\%} \text{ (Hake)}$$

While the Hake Gain helps mitigate the problem of how to compare teaching methods between schools whose students have different levels of preparation for physics, there are flaws in the statistic. Students' pretest scores are not solely based on their incoming knowledge. As both the FCI and CSEM are multiple-choice tests, very few students make a score below what they would have made by merely guessing. This project seeks to adjust the Hake Gain by taking guessing into account, and to see if the adjusted Hake Gains change the conclusions made about various physics teaching methods.

Chapter 2: DATA COLLECTION AND STATISTICAL ANALYSIS

Data Collection

University Physics I (UPI) and University Physics II (UPII) are the introductory calculus-based physics courses at the University of Arkansas. UPI covers mechanics and UPII covers electricity and magnetism. The University Physics sequence is required for most University of Arkansas students in engineering or the physical sciences. Some of the students have taken physics classes in high school, but some are seeing this material for the first time. Calculus I is a co-requisite for UPI and Calculus II is a co-requisite for UPII, so the students in UP have a wide range of levels of mathematical sophistication.

At the beginning and the end of the semester, UPI students are given the FCI and UPII students are given the CSEM as a graded test. The student's response to each question was recorded. This study uses the pretest and posttest scores from the Fall 2002, Spring 2003, Fall 2003, and Spring 2004 semesters. During those semesters 619 students took UPI and 382 took UPII. Only students who took both the pretest and posttest were used in the study.

Statistical Analysis

The corrected Hake Gain was constructed based on a statistical analysis of the distribution of pretest and posttest scores on the FCI and CSEM. The pretest and posttest data for the FCI is shown in Table I and for the CSEM in Table II.

Table I: FCI Pretest & Posttest

Score	# of pretest students	# of posttest students
0	2	0
1	0	0
2	0	0
3	3	0
4	11	0
5	20	0
6	37	2
7	60	0
8	37	5
9	42	13
10	55	15
11	48	18
12	24	21
13	37	22
14	34	36
15	23	41
16	26	29
17	22	34
18	16	28
19	23	32
20	21	40
21	7	45
22	9	33
23	6	45
24	6	33
25	6	34
26	3	27
27	6	23
28	2	14
29	3	15
30	0	14

Figure 1

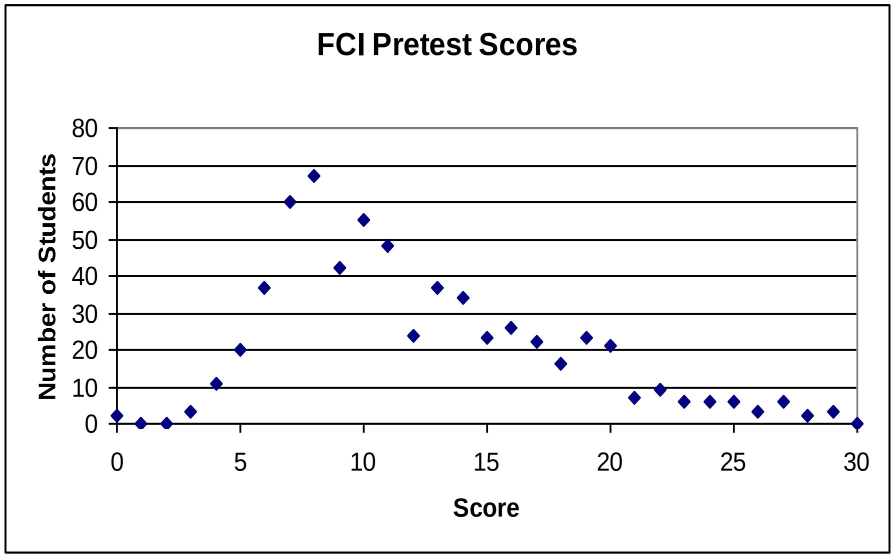


Figure 2

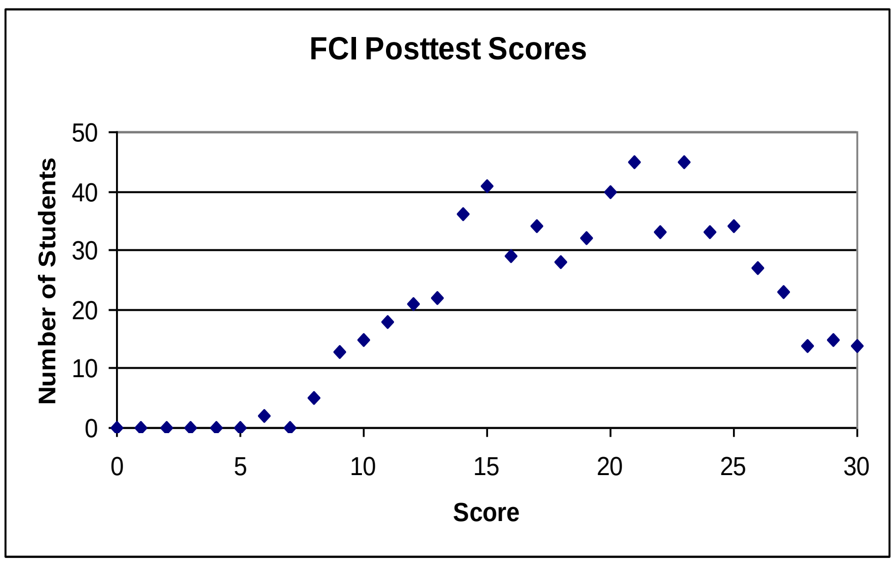


Table II: CSEM Pretest and Posttest

Score	# of pretest students w/ score	# of posttest students w/ score
0	0	0
1	0	0
2	0	0
3	2	0
4	11	0
5	17	0
6	33	1
7	54	1
8	57	2
9	47	3
10	36	11
11	42	13
12	23	20
13	20	14
14	12	19
15	7	25
16	5	24
17	5	22
18	4	18
19	1	20
20	1	30
21	2	21
22	2	25
23	1	31
24	0	22
25	0	23
26	0	12
27	0	10
28	0	9
29	0	6
30	0	0
31	0	0
32	0	0

Figure 3

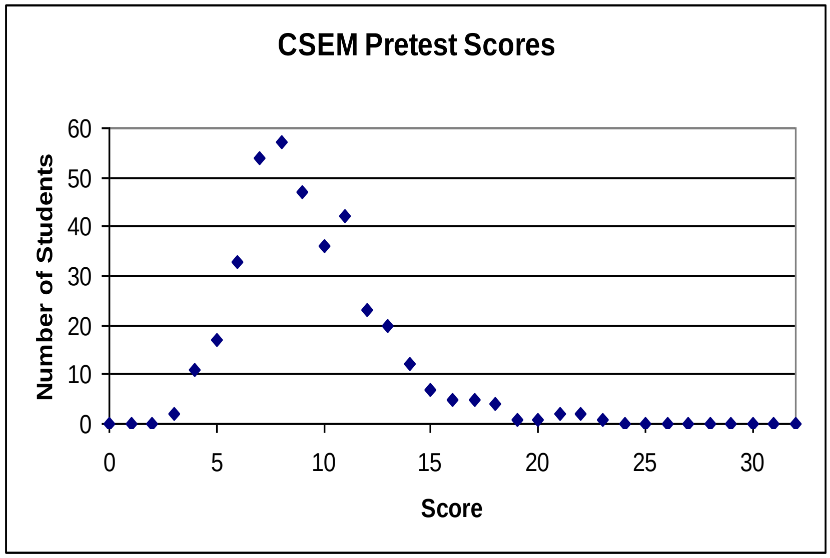
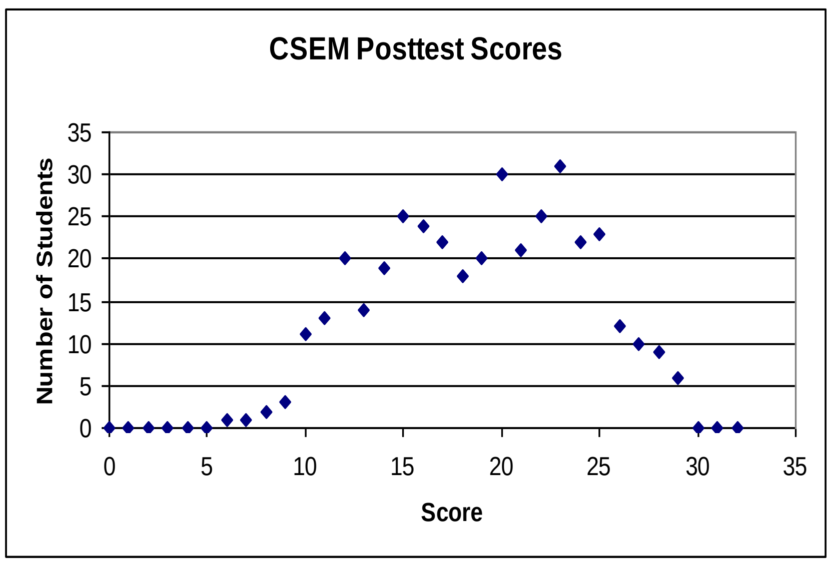


Figure 4



In order to develop a correction to the Hake Gain, the first step was to find a statistical fit for the data. Initial testing was done with the UPII pretest data. Since students have less knowledge about the subject when they start the class, it has a distribution with a simpler structure.

Initially it was thought that the data would have some sort of binomial distribution. The simplest model of the state of student knowledge for the questions contained in the FCI and CSEM is that a subset of students N_i, answer correctly with probability P_i. The sum of the subsets equals the total number of students. $N_s = N_1 + \ldots + N_m$. The responses within each subset would have a binomial distribution. The analysis that follows will determine if this model fits the data and determine the Ns and Ps for both posttest and pretest. This model is equivalent to an average student answering N_i questions with probability P_i.

The equation for the binomial distribution takes in three variables, N, the number of trials, P, the probability of success for each trial, and X, the desired number of successes. The equation for the binomial probability distribution is:

$$b(X,N,P) = \binom{N}{X} P^X (1-P)^{N-X} \text{ for x = 0, 1, ... N} \quad \text{(Freund and Walpole).}$$

For the FCI, N = 30 and for the CSEM, N = 32.

Since the binomial distribution is discrete, standard statistical fitting software, such as SAS, could not fit combinations of the distribution. The fitting algorithm in those programs takes the derivative of the equation you are trying to fit the data to, and an equation must be continuous to be differentiable. As $N \to \infty$, the binomial distribution approaches the normal distribution, which is continuous. It is defined with the equation

$$n(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

where μ is the mean and σ is the standard deviation. The binomial distribution approaches the normal distribution with $\mu = NP$ and $\sigma = \sqrt{NP(1-P)}$. If these are substituted into the normal distribution equation, it becomes a continuous approximation to the binomial distribution.

$$b_{approx}(X,N,P) = \frac{1}{\sqrt{2\pi NP(1-P)}} e^{-\frac{1}{2}\left(\frac{x-NP}{\sqrt{NP(1-P)}}\right)^2}$$

$$= \frac{1}{\sqrt{2\pi N}} \frac{\theta}{\sqrt{P(1-P)}} e^{\frac{(x-NP)^2}{2NP(1-P)}}$$

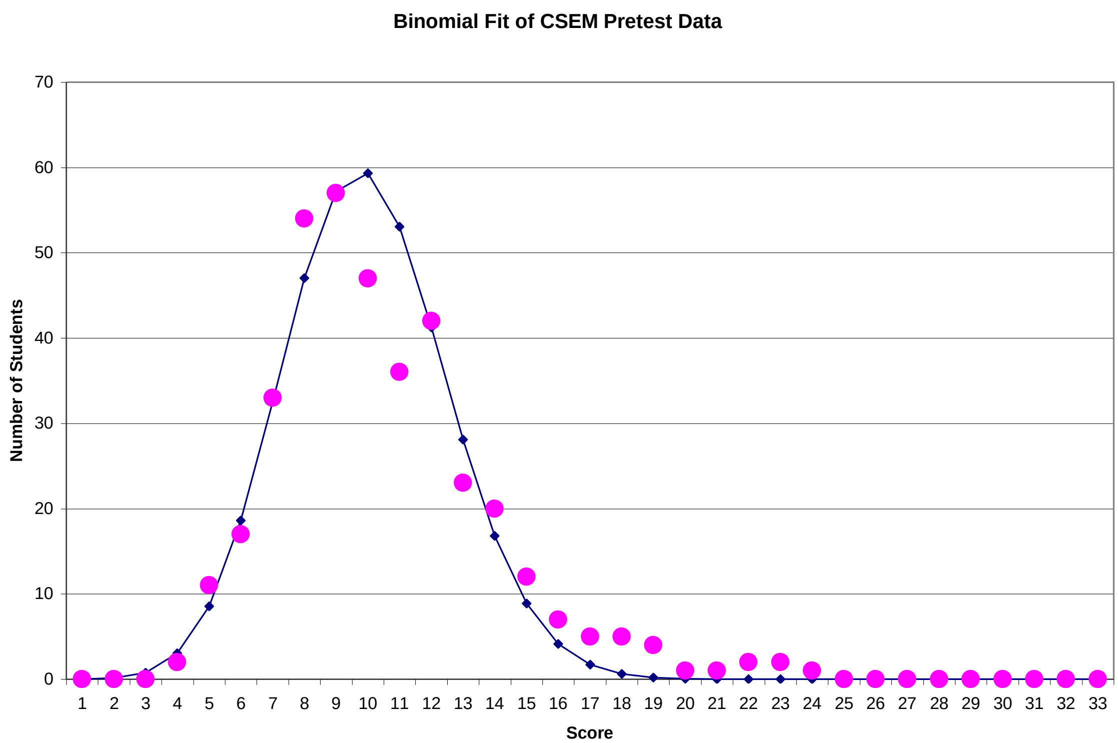
The normal distribution is a good approximation of the binomial distribution if both NP and N(1-P) are greater than 5. Solving these equations for N= 30, we find that P must be between 1/6 and 5/6 and for N = 32, P must lie between 5/32 and 27/32. So for either the FCI or the CSEM, if P is between .15 and .85, the normal distribution is a good approximation. The equation b_{approx} was tested against generated binomial data and the fit was quite good with the fitting program extracting the correct binomial parameters.

To further test the applicability of the b_{approx} equation against the real CSEM data, a program was constructed to fit the discrete, b, distribution for up to 2 subsets. The program generated a series of binomial curves for all values of P between 0 and 1, to an accuracy of 0.01. Each curve was compared to the values of the data to be fitted by taking difference between each data point from 1 to N, squaring it, and taking the sum of those. The value of P that had the smallest sum of the squares of differences was chosen as the correct fit. (See appendix for program listing) This program's role was to ensure that the normal distribution was indeed a good approximation of the binomial distribution so that the two could be used interchangeably.

Chapter 3: DATA ANALYSIS

Initially, the CSEM pretest data was modeled with a single binomial which would represent a model of student knowledge where each student had the same probability of answering each question. This fit qualitatively, but failed to capture the fine features of the distribution.

Figure 5



The binomial fit to the UPII pretest data in Figure 5 was promising, but the fit failed to capture the large tail of the graph. This indicated a model with more than one subset and probability was required. The two subset equation was :

$$b(X,N,P_1,P_2,\theta)=\theta\cdot b(X,N,P_1)+(1-\theta)\cdot b(X,N,P_2)$$

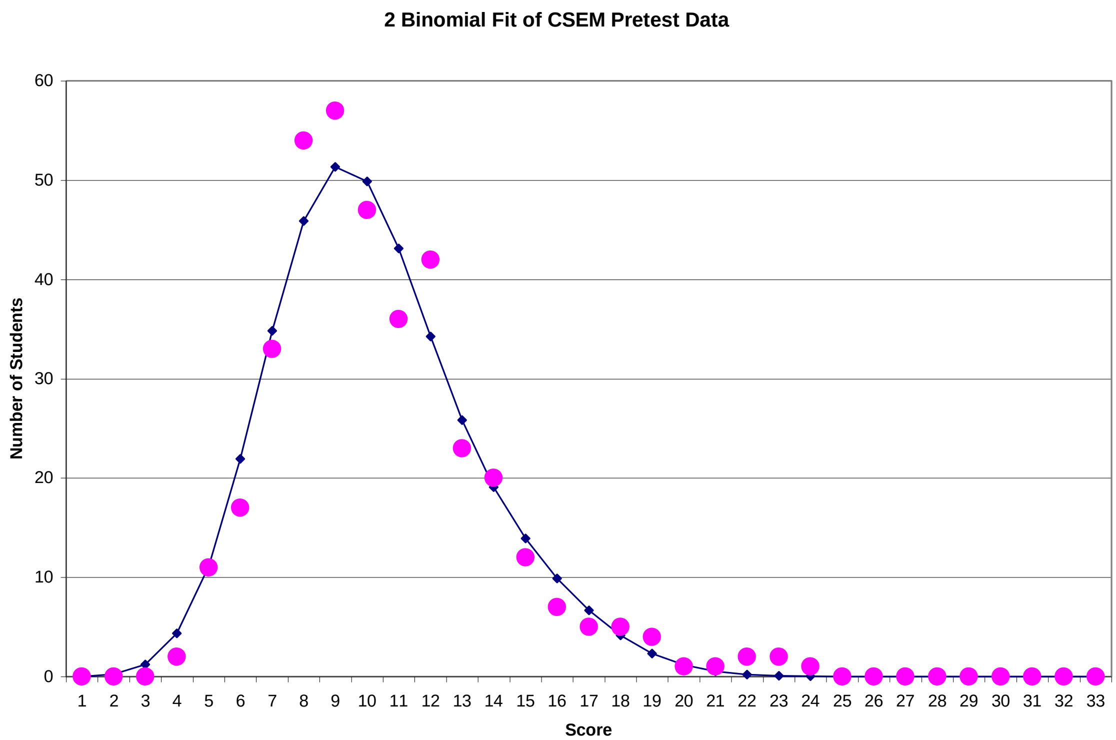
$$b_{approx}(X,N,P_1,P_2,\theta) = \theta \cdot b_{approx}(X,N,P_1) + (1-\theta)\cdot b_{approx}(X,N,P_2)$$

$$= \theta \cdot \frac{1}{\sqrt{2\pi NP_1(1-P_1)}} e^{-\frac{1}{2}\left(\frac{x-NP_1}{\sqrt{NP_1(1-P_1)}}\right)^2} + (1-\theta)\cdot \frac{1}{\sqrt{2\pi NP_2(1-P_2)}} e^{-\frac{1}{2}\left(\frac{x-NP_2}{\sqrt{NP_2(1-P_2)}}\right)^2}$$

$$= \frac{1}{\sqrt{2\pi N}} \left(\frac{\theta}{P_1(1-P_1)} e^{\frac{(x-NP_1)^2}{2NP_1(1-P_1)}} + \frac{1-\theta}{P_2(1-P_2)} e^{\frac{(x-NP_2)^2}{2NP_2(1-P_2)}} \right)$$

where P_1 and P_2 are the probabilities of the two binomial distributions and θ is a number between 0 and 1. θ is the fraction of the students in subset 1 and 1- θ is the fraction of students in subset 2. The fit using two subsets was excellent for CSEM pretest data (Figure 6). Fitting the normal approximation to the CSEM with SAS leads to almost exactly the same values for P_1, P_2, and θ that were found by the binomial fitting program.

Figure 6



Similar fits were done for the UPII posttest, UPI pretest and UPII posttest data, but those fits were not as convincing (see Figures 7-9).

Figure 7

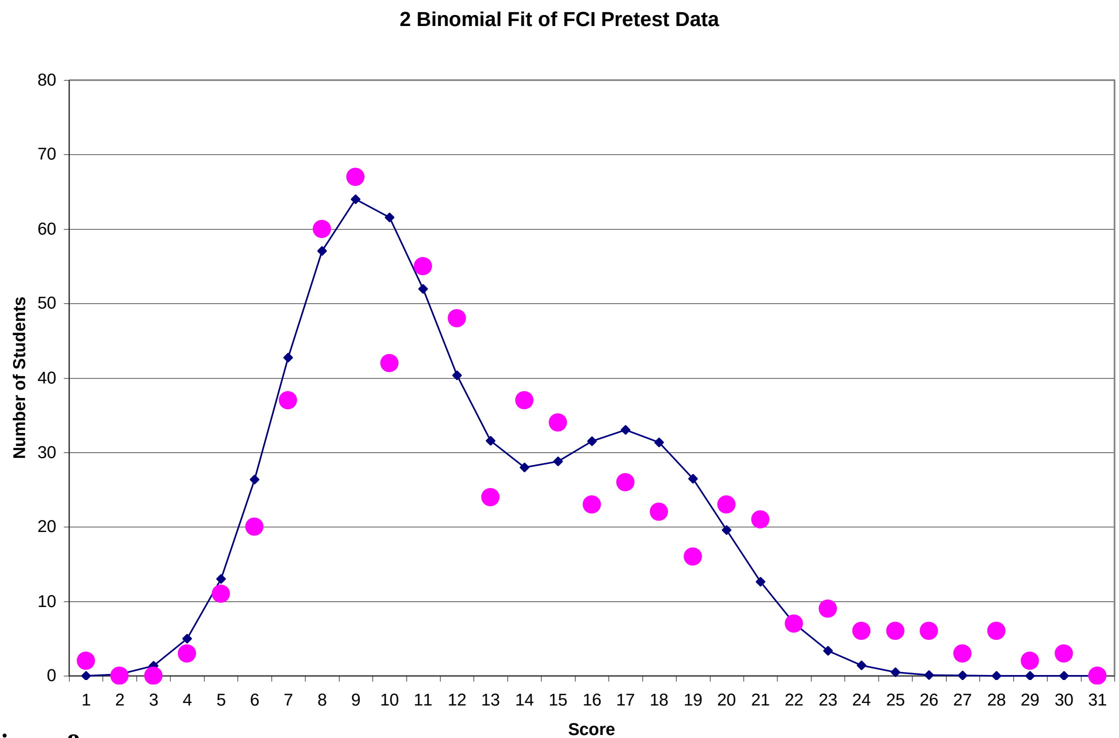


Figure 8

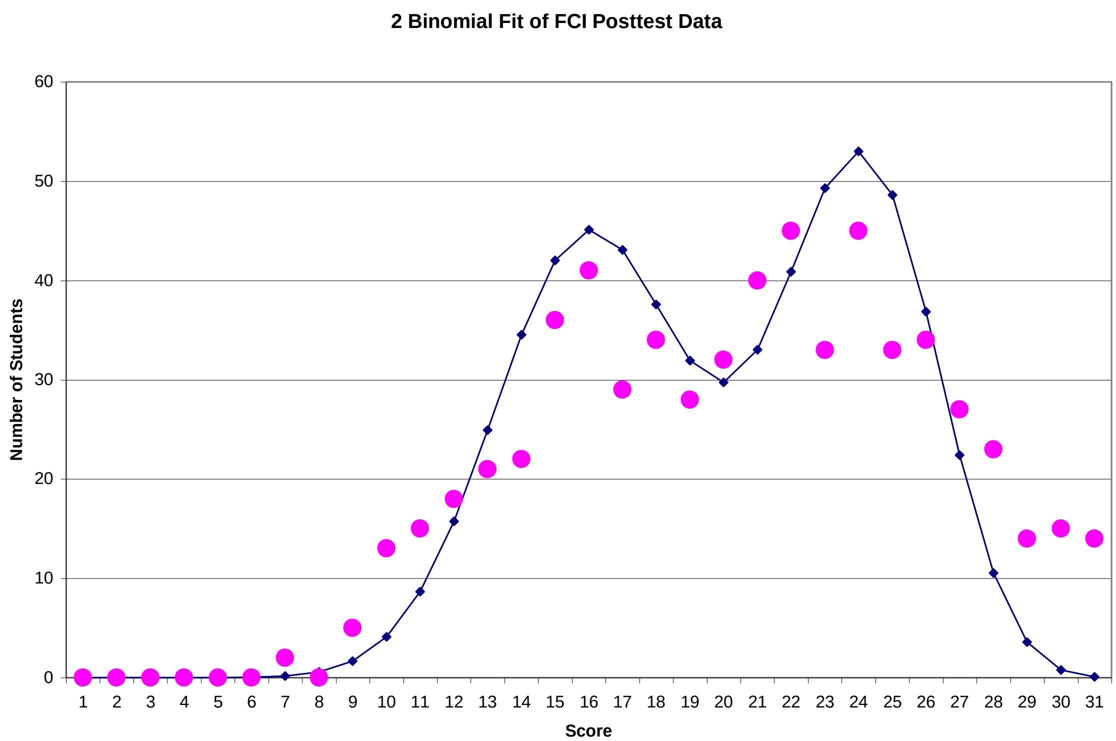
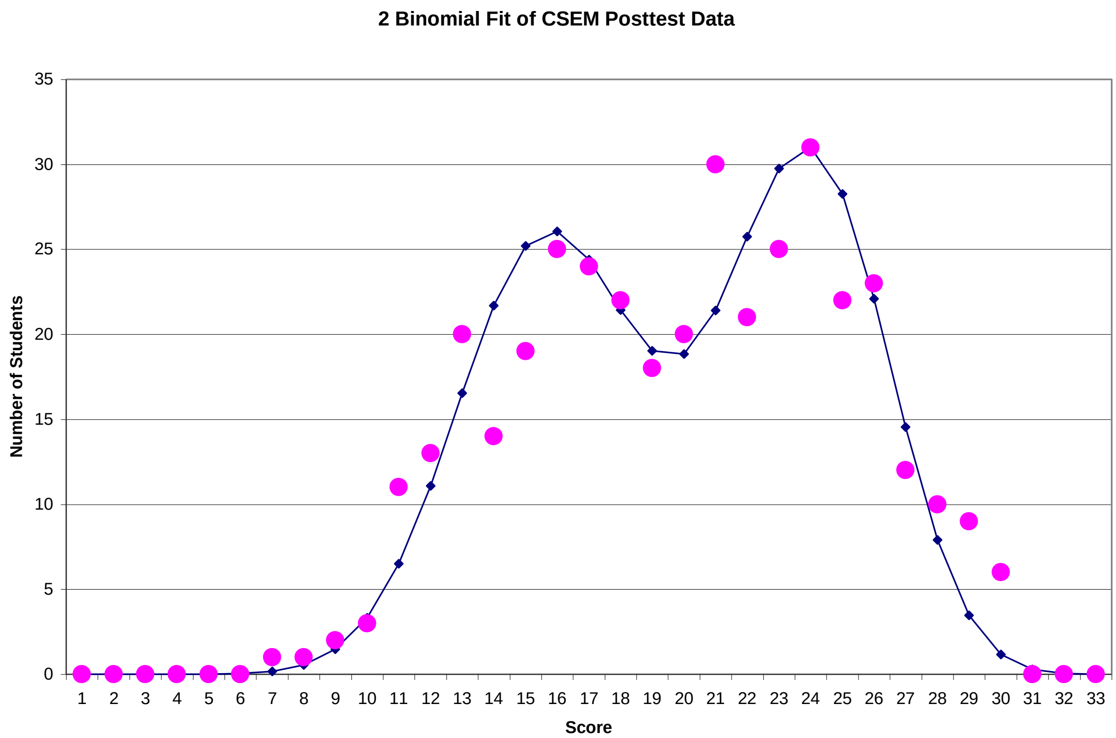


Figure 9



A model that uses two normal distributions fits the CSEM pretest data well. The next step was to attempt to improve the fit by generalizing the model to include three and four binomial distributions. The 3 binomial distributions and 4 binomial distributions are defined below.

$$b(X,N,P_1,P_2,P_3,\theta_1,\theta_2) = \theta_1 \cdot b(X,N,P_1) + \theta_2 \cdot b(X,N,P_2) + (1-\theta_1-\theta_2)\cdot b(X,N,P_3)$$

$$b(X,N,P_1,P_2,P_3,P_4,\theta_1,\theta_2,\theta_3) = \theta_1 \cdot b(X,N,P_1) + \theta_2 \cdot b(X,N,P_2) + \theta_3 \cdot b(X,N,P_3) + (1-\theta_1-\theta_2-\theta_3)\cdot b(X,N,P_4)$$

All four data sets, CSEM pretest and posttest and FCI pretest and posttest, were fit using the 2 binomial, 3 binomial and 4 binomial distributions. In three cases, the 3 binomial was the one that fit the best. In the case of the CSEM pretest data, the improvement was subtle, but noticeable. The FCI posttest data was the only that required a 4 binomial distribution fit that was significantly different from the corresponding 3 binomial distribution. For every other data set, the 4 binomial distribution either failed to fit, fit with invalid parameters ($\theta > 1$ or $\theta < 0$) or had a very low θ that made it essentially the same as the 3 binomial distribution.

Normally, one can compare the goodness of fit of different distributions by comparing the size of the F statistic for that distribution (Montgomery). However, the F statistic is dependent on the degrees of freedom of the distribution, and each of our distributions has a different number of degrees of freedom. The 2 binomial has 3 degrees of freedom, the 3 binomial has 5 and the 4 binomial has 7. After selecting the F distribution with the appropriate number of degrees of freedom it was found that the α for every fit was less than .00000001, and therefore the models are statistically indistinguishable. This statistical indistinguishability forces the choice of the model with the most subsets where fitting converged and the parameters made statistical sense. The best models are shown in Figures 10-13.

Figure 10

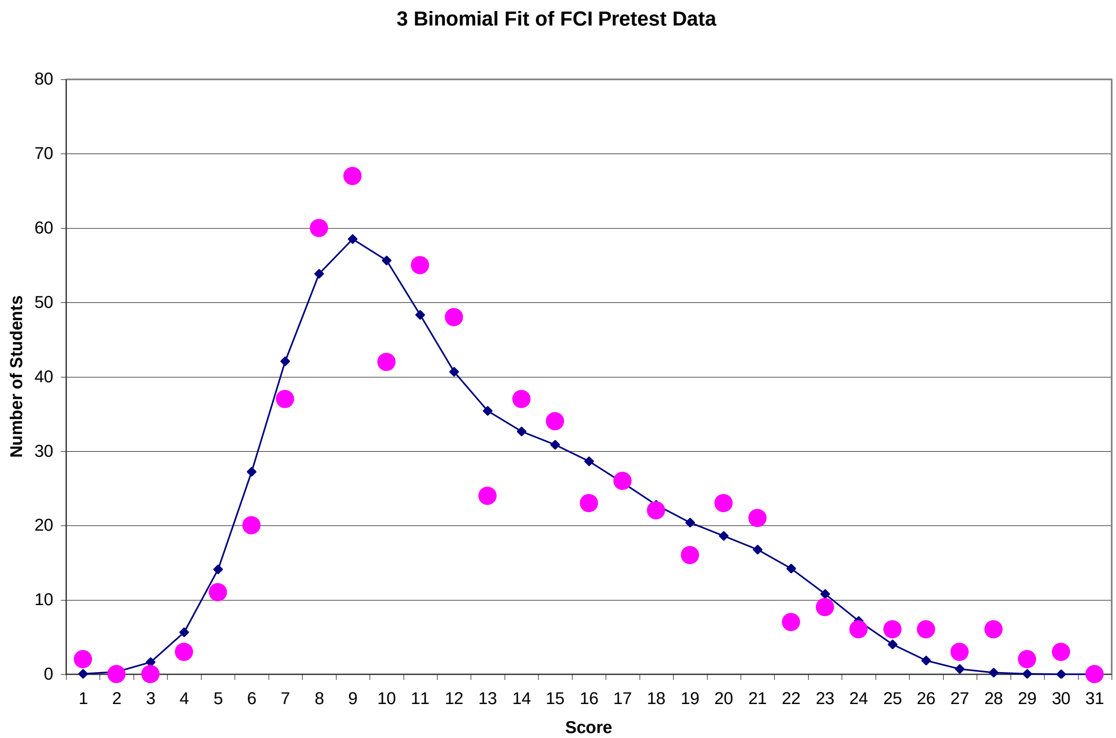


Figure 11

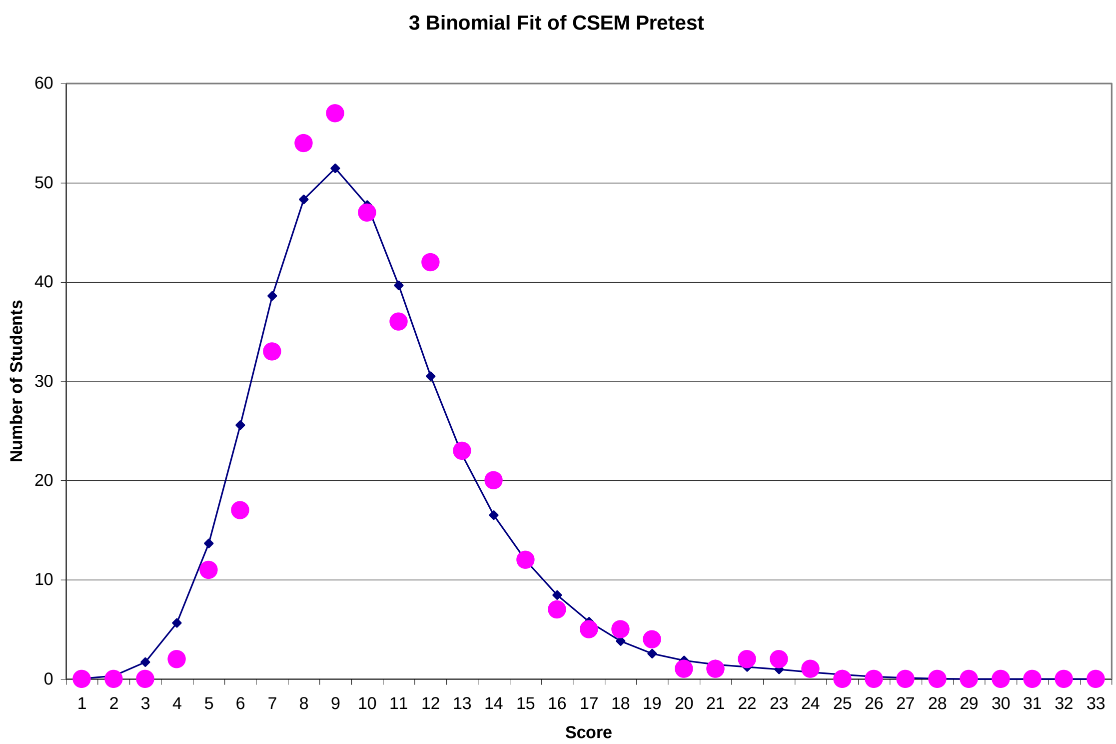


Figure 12

4 Binomial Fit of FCI Posttest Data

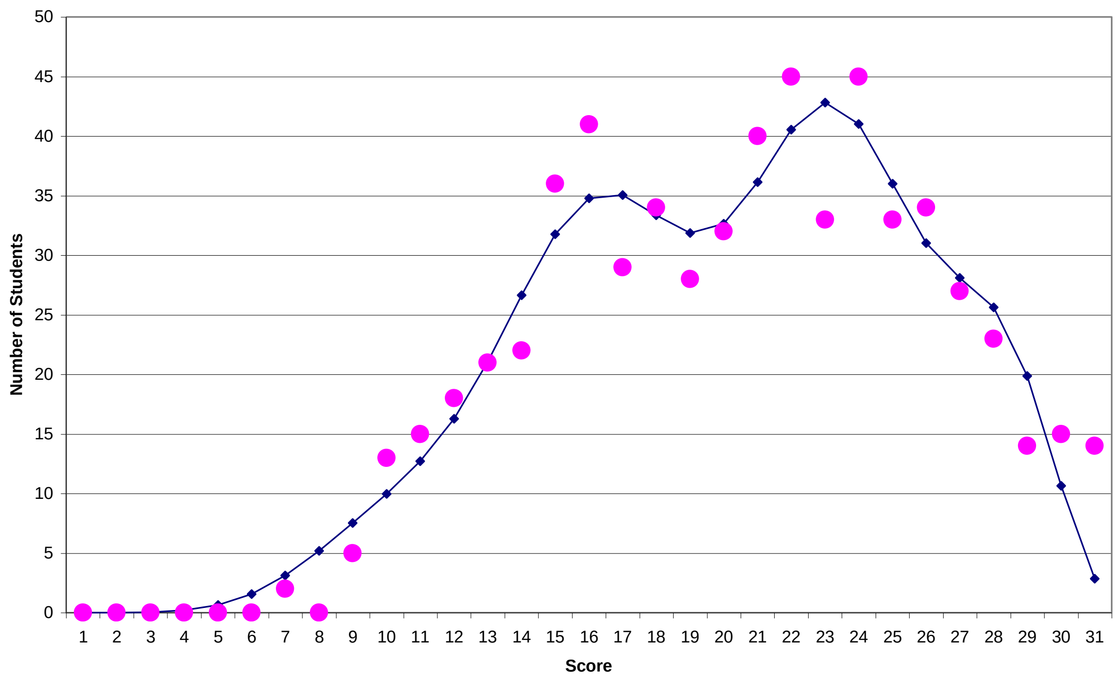
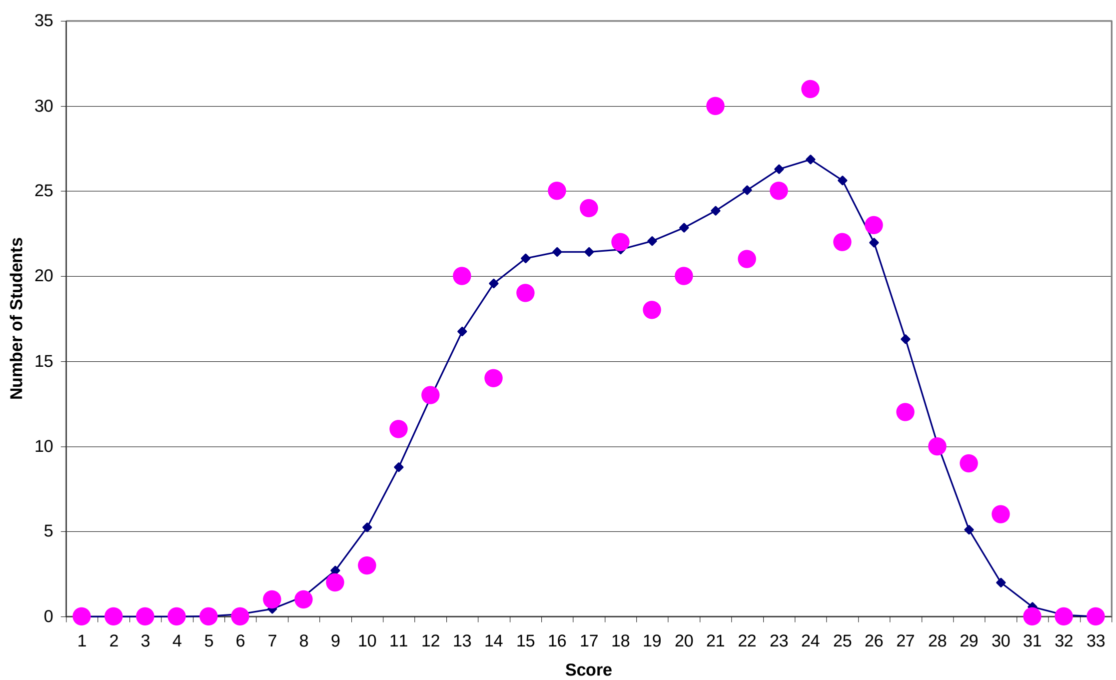


Figure 13

3 Binomial Fit for CSEM Posttest Data



The fits allow the extraction of the sizes of the subsets of students N_i and the probability of a student in that subset answering correctly P_i. The coefficient, θ, that precedes each binomial equation is a fraction of students, and the probability, P, for that binomial equation is the rate at which those students are answering questions correctly. These results are presented in Tables III-VI.

Table III: FCI Pretest

Percent of Students	Probability of Answering Question Correctly
56%	0.27
29%	0.47
15%	0.66

Table IV: CSEM Pretest

Percent of Students	Probability of Answering Question Correctly
80%	0.25
18%	0.40
2%	0.62

Table V: FCI Posttest

Percent of Students	Probability of Answering Question Correctly
8%	0.32
37%	0.51
40%	0.73
15%	0.89

Table VI: CSEM Posttest

Percent of Students	Probability of Answering Question Correctly
32%	0.42
30%	0.59
38%	0.72

On the FCI pretest, 56% of the students had answered with a 27% chance of getting it right. On the CSEM pretest, 80% answered with a similar 25%. Both of these probabilities represent a student who can only eliminate one of the five multiple choice answers and therefore represent students with very little facility with the problem and are mostly guessing. On both posttests, the percent of answering at a rate consistent with mere guessing is negligible.

Chapter 3: CORRECTING THE HAKE GAIN

The score on a multiple choice test is an estimate of a student's knowledge. That estimate is biased when the student is presented with a multiple choice instrument covering material of which they have little knowledge. Since a student with no knowledge should score 20% by mere chance when given 5 choices, this inflates the pretest scores and biases the Hake Gain toward lower values. The statistical models presented above can be used to create and unbiased estimate of the pretest score. The pretest & posttest score can be calculated from our model by taking a weighted (by percent of student) sum of the different answer rates.

$$pretest = \sum \frac{N_i}{N} P_i$$

These calculated percentages agree very well with the percentages calculated directly from the data. This confirms the goodness of fit of the models. In order to find an effective pretest score, the portion subset of the students that are correctly answering at a rate consistent with guessing is removed.

$$effectivepretest = \sum \frac{N_i}{N} P_i, P_i \geq .30$$

The corrected conceptual gain is constructed by replacing the pretest score in the Hake Gain equation with the effective pretest score, and the resulting equation is the Corrected Hake Gain.

$$CorrectedHakeGain \equiv \frac{(posttest)\% - (effectivepretest)\%}{100\% - (effectivepretest)\%}$$

Table VII: Original and Effective Pretest and Posttest Scores

	Original	Effective
FCI pretest	38.7%	23.6%
FCI posttest	64.0%	-
CSEM pretest	28.4%	8.4%
CSEM posttest	58.5%	-

Table VIII: Original and Effective Hake Gains

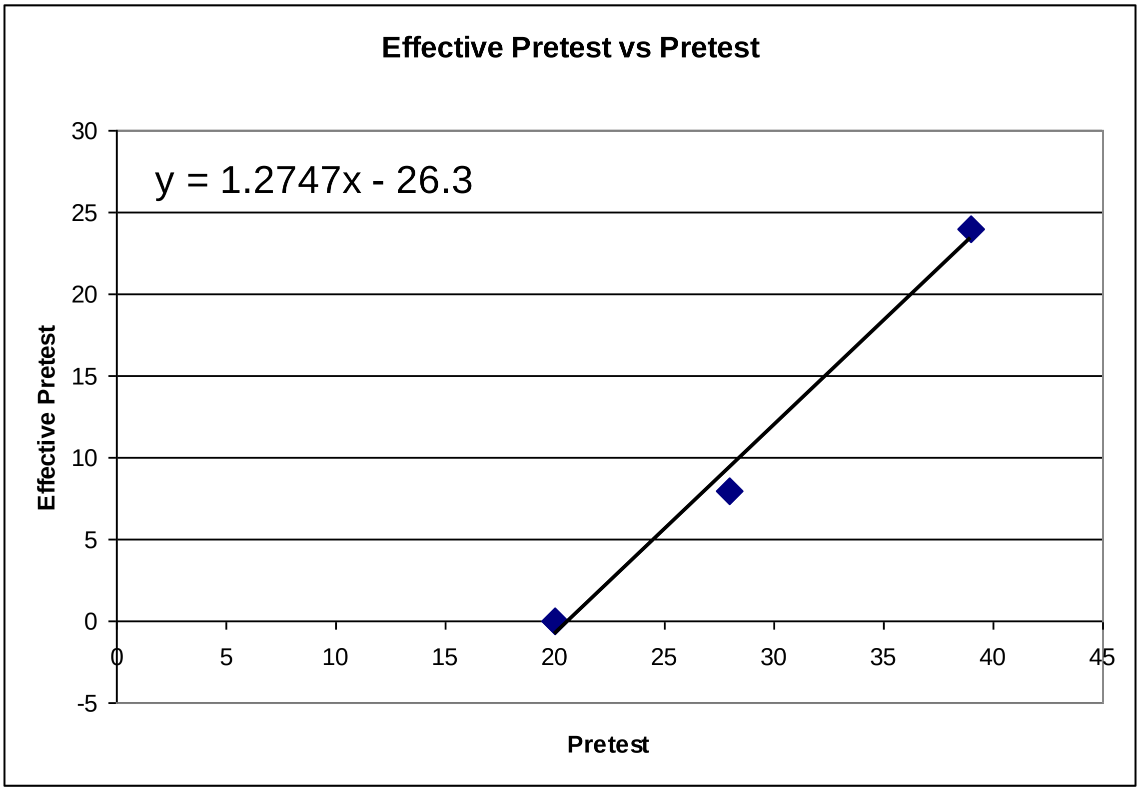
	Hake Gain	Corrected Hake Gain
FCI	41.3%	52.8
CSEM	42%	54.7

An equation for correcting the pretest score of a class, when per student data is unavailable, can be constructed using the correction for the two classes and the observation that a pretest score of 20% represents an effective pretest score of zero.

Table IX: Data Used to Construct Effective Pretest

pretest	corrected
20	0
28	8
39	24

Figure 14



Fitting a line to the effective pretest score as shown in Figure 14 yields the following equation for the effective pretest score.

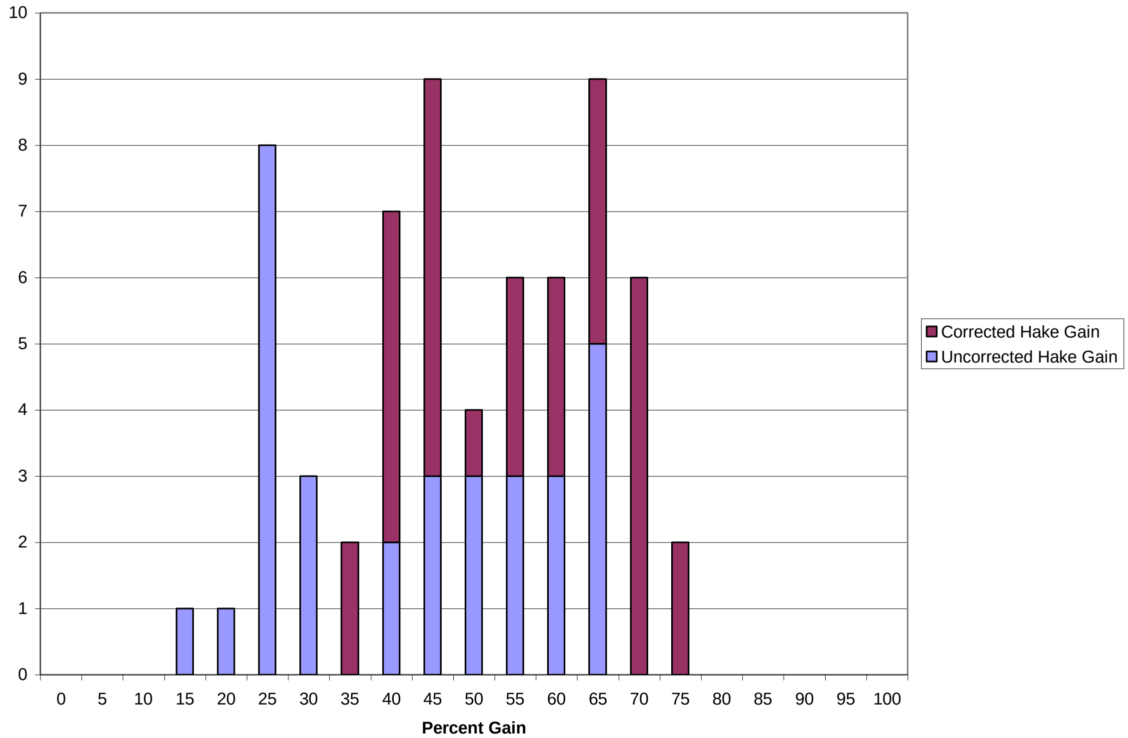
$$effectivepretest \approx 1.27 \cdot pretest - 26.3$$

Hake separated classes into three categories, Low Gain (Hake Gain ≤ 0.3), Medium Gain ($0.3 \leq$ Hake Gain ≤ 0.7) and High Gain (Hake Gain ≥ 0.7). The Lower Gain courses used traditional instruction, whereas the others used interactive methods. Removing the bias due to guessing increases the Hake Gain, but how does that affect the difference in Hake Gains between interactive and traditionally taught classes? Table X and Figure 15 show the effect of applying the Corrected Hake Gain to university science classes at a variety of institutions and in a variety of settings. (Hake)

Table X: Results of Correcting Hake Gain

Course	Type	Pretest	Posttest	Eff. Pretest	Hake Gain	Eff. Hake Gain
ASU	Traditional	37	53	20.69	25.40	40.74
ASU1	Traditional	52	64	39.74	25.00	40.26
ASU2	Traditional	52	63	39.74	22.92	38.60
ASU-HH	Interactive	48	75	34.66	51.92	61.74
ASU-MW	Interactive	36	68	19.42	50.00	60.29
ASU-VH	Interactive	34	63	16.88	43.94	55.49
CP	Traditional	44	58	29.58	25.00	40.36
CP-RK	Interactive	59	84	48.63	60.98	68.85
CP-RK-A	Interactive	46	72	32.12	48.15	58.75
CP-RK-B	Interactive	55	81	43.55	57.78	66.34
EM-90	Traditional	70	78	62.60	26.67	41.18
EM-91	Interactive	71	85	63.87	48.28	58.48
EM-93	Interactive	70	86	62.60	53.33	62.57
EM-94	Interactive	71	88	63.87	58.62	66.79
EM-95	Interactive	67	88	58.79	63.64	70.88
IUPRE93	Interactive	44	74	29.58	53.57	63.08
IU93S	Interactive	37	73	20.69	57.14	65.96
IU94S	Interactive	40	79	24.50	65.00	72.19
IU95S	Interactive	42	77	27.04	60.34	68.48
IU95F	Interactive	32	74	14.34	61.76	69.65
MICH-DE	Interactive	42	67	27.04	43.10	54.77
MICH1	Interactive	47	67	33.39	37.74	50.46
MICH2	Interactive	45	65	30.85	36.36	49.39
MICH3	Interactive	39	53	23.23	22.95	38.78
MICH4	Interactive	31	47	13.07	23.19	39.03
OS92	Traditional	48	55	34.66	13.46	31.13
OS95	Interactive	48	70	34.66	42.31	54.09
UL94F	Traditional	44	54	29.58	17.86	34.68
UL-RM95S	Interactive	43	58	28.31	26.32	41.41
UL-RM95SU	Interactive	44	58	29.58	25.00	40.36
UML93	Interactive	40	54	24.50	23.33	39.07
UML94	Interactive	38	51	21.96	20.97	37.21

Figure 15: Hake Gains at Multiple Institutions



While the corrected Hake Gain does reduce the gap between interactive and traditional classes, the interactive classes still produce higher gains. This shifts the perception of the efficacy of interactive engagement from a vital solution to a crisis to merely an effective improvement.

Chapter 6: CONCLUSIONS

Pretest and posttest scores for the FCI and CSEM can be easily modeled using a sum of binomial or normal distributions. By removing the part of the distribution that corresponds to a rate close to that of random guessing, an effective pretest score that is approximately what students would achieve on a non-multiple choice test can be found. The effect of the correction is to move the types of courses closer together, but interactive classes still produce higher gains.

Bibliography

Crouch, Catherine H. and Eric Mazur, "Peer Instruction: Ten years of experience and results," *American Journal of Physics* 70, No. 12, (2002)

Freund, John and Ronald Walpole, Mathematical Statistics, 3rd Ed. Prentice-Hall Inc. 1980

Hake, R.R., "Interactive-engagement vs. traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses," *American Journal of Physics* 66, 64-74 (1998).

Hake, R.R., "Interactive-engagement methods in introductory mechanics courses," Preprint (1998)

Hestenes, David, Gregg Swackhamer, and Malcolm Wells. "Force Concept Inventory." The Physics Teacher, 30 (March 1992), 141-158.

Maloney, David P., Thomas L. O'Kuma, Curtis J. Hieggelke, Alan Van Heuvelen, "Surveying students' conceptual knowledge of electricity and magnetism," *Physics Education Res., American Journal of Physics Suppl.*, 69, No. 7 (2001).

Montgomery, Douglas C., Elizabeth A. Peck, G. Geoffrey Vining, Introduction to Linear Regression Analysis, 3rd Ed. Wiley Interscience 2001.

National Science Board, "Science and Technology: Public Attitudes and Public Understanding," Science and Engineering Indicators 2000, Vol.1 (2000).

"SAS OnlineDoc V8" http://www.ncsu.edu/it/sas/help/.

Appendix One

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>


float numStudents = 0;
int numQuestions = 0; //30 for FCI, 32 for CSEM

double realData[33];

void main()
{
    float binGenerate(float);
    float binGeneratePrint(float);

    void dataSetup(); //sets up data

    float bestDiff = 100000;
    float thisDiff;
    float bestTheta;

    float testTheta = (float)0.01;

    dataSetup();

    while(testTheta <=1)
    {
        thisDiff = binGenerate(testTheta);
        if(thisDiff < bestDiff)
        {
            bestDiff = thisDiff;
            bestTheta = testTheta;
        }

        testTheta += (float)0.01;
    }


    printf("the best fit is Theta = %f \n", bestTheta);
    binGeneratePrint(bestTheta);

}
```

```
float binGenerate(float Theta)
{
    float Combin(int, int);

    int N = numQuestions;
    float totals[33]; // curve
    float differences[33];
    float avediffsq = 0;
    float temp1 = 0;
    float temp2 = 0;
    float temp3 = 0;
    int i = 0;

    for(i = 0; i<=N; i++)
    {
            temp1 = Combin(N, i);
            temp2 = (float)pow(Theta, i);
            temp3 = (float)pow((1 - Theta), (N - i));
            totals[i] = temp1*temp2*temp3*numStudents; //generate  curve

    }

    for (i = 0; i<N + 1; i++)
    {
            differences[i] = (float)(((float)totals[i] - (float)realData[i])*((float)totals[i]
    - (float)realData[i]));
            avediffsq += differences[i];
    }

    avediffsq = avediffsq/(numQuestions +1);

    return avediffsq;
}

float binGeneratePrint(float Theta)
{
    float Combin(int, int);

    int N = numQuestions;
    float totals[33]; // curve
    float temp1 = 0;
    float temp2 = 0;
    float temp3 = 0;
    int i = 0;
    FILE *fpointer = NULL;
```

```
    char filename[30];

    for(i = 0; i<=N; i++)
    {
            temp1 = Combin(N, i);
            temp2 = (float)pow(Theta, i);
            temp3 = (float)pow((1 - Theta), (N - i));
            totals[i] = temp1*temp2*temp3*numStudents; //generate  curve
    }

    sprintf(filename, "totals.%f.txt", Theta);
    fpointer = fopen(filename, "w+");

    for (i = 0; i<numQuestions + 1; i++)
    {
            fprintf(fpointer, "%f\n", totals[i]);
    }

    return 0;
}

float Combin(int N, int K)
{
    float temp1 = 1;
    float temp2 = 1;
    float temp3 = 1;
    int i = 0;

    for(i = 1; i<=N; i++) // compute N!
    {
            temp1 *= i;
    }

    for(i = 1; i<=K; i++) // compute K!
    {
            temp2 *= i;
    }

    for(i = 1; i<=(N-K); i++) // compute (N-K)!
    {
            temp3 *= i;
    }

    return (temp1/(temp2*temp3));
}
```

```
void dataSetup()
{
    float totalStudents = 0;
    float normFactor = 0;
    int i=0;

    // Data initialization, excised for space

    //Normalization of data

    for(i=0; i <= numQuestions; i++)
    {
            totalStudents += (float)realData[i];
            //printf("%d: %f\n", i, realData[i]);
    }

    numStudents = totalStudents;


}
```

Appendix Two

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>


float numStudents = 0;
int numQuestions = 0; //30 for FCI, 32 for CSEM

double realData[33];

void main()
{
    float binGenerate(float, float, float);
    float binGeneratePrint(float, float, float);

    void dataSetup(); //sets up data

    float bestDiff = 100000;
    float thisDiff;
    float bestP;
    float bestTheta1;
    float bestTheta2;

    float testP = (float)0.01;
    float testTheta1 = (float)0.01;
    float testTheta2 = (float)0.01;

    dataSetup();

    while (testP <= 1)
    {
            //printf("%f\n", testP);
            testTheta1 = 0;
            while(testTheta1 <=1)
            {
                    testTheta2 = 0;
                    while(testTheta2 <=1)
                    {
                            thisDiff = binGenerate(testP, testTheta1, testTheta2);
                            if(thisDiff < bestDiff)
                            {
                                    bestDiff = thisDiff;
                                    bestP = testP;
                                    bestTheta1 = testTheta1;
```

```
                    bestTheta2 = testTheta2;
                }

                testTheta2 += (float)0.01;
            }

            testTheta1 += (float)0.01;
        }
        testP += (float)0.01;
    }

    printf("the best fit is P = %f, Theta1 = %f, Theta2 = %f \n", bestP, bestTheta1,
    bestTheta2);
    binGeneratePrint(bestP, bestTheta1, bestTheta2);

}


float binGenerate(float P, float Theta1, float Theta2)
{
    float Combin(int, int);

    int N = numQuestions;
    float totals1[33]; //Theta1 curve
    float totals2[33]; //Theta2 curve
    float totals[33]; //combined curve
    float differences[33];
    float avediffsq = 0;
    float temp1 = 0;
    float temp2 = 0;
    float temp3 = 0;
    int i = 0;

    for(i = 0; i<=N; i++)
    {
        temp1 = Combin(N, i);
        temp2 = (float)pow(Theta1, i);
        temp3 = (float)pow((1 - Theta1), (N - i));
        totals1[i] = temp1*temp2*temp3*numStudents; //generate theta1 curve

        temp2 = (float)pow(Theta2, i);
        temp3 = (float)pow((1 - Theta2), (N - i));
        totals2[i] = temp1*temp2*temp3*numStudents; //generate theta2 curve

        totals[i] = (1 - P)*totals1[i] + P*totals2[i];
    }
```

```
	for (i = 0; i<N + 1; i++)
	{
		differences[i] = (float)(((float)totals[i] - (float)realData[i])*((float)totals[i]
	- (float)realData[i]));
		avediffsq += differences[i];
	}

	avediffsq = avediffsq/(numQuestions +1);

	return avediffsq;
}

float binGeneratePrint(float P, float Theta1, float Theta2)
{
	float Combin(int, int);

	int N = numQuestions;
	float totals1[33]; //Theta1 curve
	float totals2[33]; //Theta2 curve
	float totals[33]; //combined curve
	float temp1 = 0;
	float temp2 = 0;
	float temp3 = 0;
	int i = 0;
	FILE *fpointer = NULL;
	char filename[30];

	for(i = 0; i<=N; i++)
	{
		temp1 = Combin(N, i);
		temp2 = (float)pow(Theta1, i);
		temp3 = (float)pow((1 - Theta1), (N - i));
		totals1[i] = temp1*temp2*temp3*numStudents; //generate theta1 curve

		temp2 = (float)pow(Theta2, i);
		temp3 = (float)pow((1 - Theta2), (N - i));
		totals2[i] = temp1*temp2*temp3*numStudents; //generate theta2 curve

		totals[i] = (1 - P)*totals1[i] + P*totals2[i];
	}

	sprintf(filename, "totals.%f.%f.%f.txt", P, Theta1, Theta2);
	fpointer = fopen(filename, "w+");

	for (i = 0; i<numQuestions + 1; i++)
```

```
    {
        fprintf(fpointer, "%f\n", totals[i]);
    }

    return 0;
}

float Combin(int N, int K)
{
    float temp1 = 1;
    float temp2 = 1;
    float temp3 = 1;
    int i = 0;

    for(i = 1; i<=N; i++) // compute N!
    {
        temp1 *= i;
    }

    for(i = 1; i<=K; i++) // compute K!
    {
        temp2 *= i;
    }

    for(i = 1; i<=(N-K); i++) // compute (N-K)!
    {
        temp3 *= i;
    }

    return (temp1/(temp2*temp3));
}


void dataSetup()
{
    float totalStudents = 0;
    float normFactor = 0;
    int i=0;

    //data initialization excised for space


    //Normalization of data

    for(i=0; i <= numQuestions; i++)
    {
```

```
        totalStudents += (float)realData[i];
        //printf("%d: %f\n", i, realData[i]);
    }

    numStudents = totalStudents;


}
```