Statistical inference in the media of standardized test scores

Leonard Sheehy

Statistics for Educational Research

New Jersey City University

The Newark Star Ledger, a daily newspaper in northern New Jersey, published an article about test results on October 20, 2015. The headline of the story was "PARCC scores: Most NJ students below grade level in Math, English" (Clark, 2015). This article discussed results of the Partnership for Assessment of Readiness for College and Careers (PARCC). PARCC is a group of states and the District of Columbia that administers standardized K–12 assessments in Mathematics and English (http://www.parcconline.org/about**)**. The PARCC exams are aligned to the Common Core curriculum standards that highlight critical thinking and problem solving skills (http://www.parcconline.org/about**)**. According to Clark (2015) any student that scores a four or five in the PARCC test is identified as meeting expectations and to be achieving at their grade level (p. 1). Clark (2015) further states that students who score a three are considered to be approaching expectations and are 'on track' (p. 1). The newspaper article stated that most of New Jersey students in grades three through eleven did not meet expectations on the PARCC exam in mathematics and English (Clark, 2015). The article goes on by comparing standardized tests administered in previous years to students in the state called the New Jersey Assessment of Skills and Knowledge and the High School Proficiency Assessment. It cites that New Jersey students scored lower on PARCC than they did on these state tests in the past (Clark, 2015). The commentary concludes that most students in other states also missed the levels for success established by the PARCC test. A conclusion is quoted in the piece by David Hespe, state Education Commissioner, declaring the results show that high school graduation requirements are not sufficiently demanding enough for the majority of children to be effective in careers after completing their education (Clark, 2015).

The newspaper cites a deduction about student's abilities without using the entities included in an inferential manner to substantiate such a conclusion. Inferential statistics employ a random sample of data obtained from a population to describe and make inferences about the population (Salkind, 2014). According to Salkind (2014) when conclusions are developed using statistics, a designed experiment that applies a treatment to individuals and attempts to separate the consequences of a treatment on a response variable should be included (p. 9). The Star Ledger fails to mention how the PARCC test accounts for other factors such as social economic ranks or nutrition levels. Therefore the test does not meet the standards of inferential studies when drawing conclusions. The PARCC test did not implement the necessary resources while their test was administered so the Star Ledger should refrain from stating that high school graduation requirements are not sufficiently demanding enough for the majority of children to be effective in careers after completing their education.

An important element included in inferential studies is the p-value. This is because for a study to prove that a claim, known as the alternative hypothesis, is true it must show convincing evidence. The null hypothesis is in opposition to the alternative hypothesis in that it represents no relationship between the variables being studied (Salkind, 2014). A p-value quantifies how likely it is to get a figure at least as significant as the observed value by random chance. The outcomes of a simulation afford an approximation of the p-value. The p-value provides readers with a quantity that measures how likely the outcome of a study could happen by incident and not the factors being studied (Tabor & Franklin, 2011). P-values fall into three categories. A quantity less than 0.01 is considered very small where random chance is not a plausible explanation for the phenomenon being tested and the null hypothesis is rejected. If the p-value is between 0.01 and 0.05 it is considered small, random chance is not plausible, and the null hypothesis is also rejected. When the p-value is greater than 0.05 random chance is a valid explanation for the results and the null hypothesis to not rejected. This leads the researcher to estimate a p-value of the PARCC tests cited in the Star Ledger to be greater than 0.05. Therefore since the p-value is not provided and the estimate is greater than 0.05 the researcher believes that more information is needed to properly evaluate statements made about PARCC test results.

In contrast to the Star Ledger article, “Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education” by Fagin and Merkle in 2002 provides details about the process on which they form their conclusions. The study measures the outcomes of a year-long trial in the use of robots in computer science education (Fagin & Merkle 2002). Freeman (2008, p. 23) states that the impact of robotic programs should be clearly stated and can be placed into categories. The impacts can relate to factual knowledge, skills, attitudes or behavior (Freeman, 2008). The Fagin and Merkle study openly defines their goal of measuring the impacts of robotics on computer programming knowledge by stating that students test performance on variables, constants, sequential control flow, and procedures without parameters is measured (Fagin & Merkle 2002). The experiment considered the results of eight hundred students on identical pre and post-tests from both robotics and non-robotics based instruction. This technique follows other education assessment methods that have been defined by Stoner (1996) and Cohen et al. (2007). A questionnaire investigating the effectiveness of robots in motivating students to study computer science or computer engineering during the semester was also conducted. This practice also is in alignment with other education valuation approaches that have been listed by Stoner (1996) and Cohen et al. (2007). The last measurement of the Fagin and Merkle study determined the effect of being in a robotics program has on a student’s probability of choosing computer science or computer engineering as a major (Fagin & Merkle 2002). The report listed the FisherIrwin test as the instrument used for this task. The Fisher's exact test of independence is implemented when two nominal variables exist and it is to be determined if the proportions of one variable are different depending on the value of the other variable. This method is widely used and supported by McDonald (McDonald, 2015). The Fagin and Merkle (2002) report provides information on the procedures followed, variables controlled and how the p-value was calculated. Tables depicting how figures were collected and processed were also given which makes it easier for the reader to evaluate and understand the information.

A comparison of these two papers demonstrates the importance of statistics when forming conclusions. Responsible researchers should engage in activities that validate their findings. When the robotics team in the second paper invested the time and effort into researching the probability of random chance in their study they fulfilled this obligation as should all researchers. This is important to not only protect the integrity of the researcher's reputation but to shelter the subjects of the study from negative consequences due to faulty research methods.

References

Clark, A. (2015, October, 15). PARCC scores: Most NJ students below grade level in math, English. *The Star Ledger*. Retrieved from nj.com on February 13, 2016.

Cohen, L., Manion, L., & Morrison, K. (2007). *Research methods in education*, New York, NY: Routledge.

Fagin, B. S., & Merkle, L. (2002). Quantitative analysis of the effects of robots on introductory Computer Science education. *Journal on Educational Resources in Computing (JERIC)*, *2*(4), 2.

Freeman, A. (Ed.). (2008). Framework for evaluating impacts of informamal science education projects. Retrieved from http://insci.org/resources/Eval_Framework.pdf

Heyboer, K. (2015, October, 21). *What do N.J.'s PARCC results mean? 5 lessons learned. The Star Ledger*. Retrieved from nj.com on February 13, 2016.

McDonald, J.H. 2014. Handbook of Biological Statistics (3rd ed.). Sparky House Publishing, Baltimore, Maryland. This web page contains the content of pages 77-85 in the printed version. Retrieved on March 9, 2016 from http://www.biostathandbook.com/fishers.html.

Owl (2015). Purdue online writing lab., retrieved from https://owl.english.purdue.edu/owl/resource/672/1/ on February 13, 2016.

Partnership for assessment of readiness for college and careers. (2015). Working together to create a modern assessment. Retrieved on February 13, 2016 from http://www.parcconline.org/about**.**

Salkind, N (2014). Statistics for People Who (Think They) Hate Statistics, N. J. New York Sage.

Sullivan, M. (2011). Statistics: Informed Decisions Using Data, Fourth Edition retrieved from https://faculty.elgin.edu/dkernler/statistics/ch01/1-2.html on February 13, 2016.

Stoner, G. (1996). *Implementing learning technology*. Heriot-Watt University, Edinburgh, UK: Learning Technology Dissemination Initiative.

Tabor, J. & Franklin, C. (2011). Statistical Reasoning in Sports. BFW publishers, New York.

PARCC scores: Most NJ students below grade level in math, English

By Adam Clark | NJ Advance Media for NJ.com

Email the author | Follow on Twitter

on October 20, 2015 at 3:09 PM, updated October 20, 2015 at 6:15 PM

TRENTON — The majority of New Jersey students in grades 3 through 11 failed to meet grade-level expectations on controversial math and English tests the state says provide the most accurate measurement of student performance yet.

New Jersey on Tuesday released the preliminary statewide results of the new online tests, the Partnership for Assessment of Readiness for College and Careers exams, revealing that more than half of the students in every grade level failed to meet expectations in math.

Students who took the test last spring also struggled to meet the benchmarks set in English, with no more than 52 percent of students in any grade level meeting expectations, which were set by the consortium of states that participated in PARCC.

New Jersey's results are similar to the other PARCC states that already released results and also saw most students miss the benchmarks for achievement.

Students receive a score on a scale of one to five. Those receiving a score of 4 or 5 are considered to be performing at their grade level and thereby meeting expectations.

Students who receive a 3 are considered to be approaching expectations, and state officials said many of those students may actually be on track, despite their score.

Overall, the results show that high school graduation requirements are not rigorous enough for most students to be successful after graduation, state Education Commissioner David Hespe said. The 2014-15 results set a new baseline for improving student achievement, he said.

"There is still much work to be done in ensuring all of our students are fully prepared for the 21st century demands of college and career," Hespe said.

The scores were announced at New Jersey Manufacturers Insurance Company in Ewing, and representatives from New Jersey colleges spoke about the importance of preparing students for college and beyond.

But state officials also stressed that scores are typically low when a new test is given in the first year. Students do not need to meet expectations on PARCC in order to move to the next grade level or graduate.

"Ultimately, PARCC is just a test," Hespe said.

RELATED: Test yourself: Can you answer these PARCC questions?

The PARCC exams are aligned to the Common Core curriculum standards, which emphasize problem solving and critical thinking. The exams, which debuted in New Jersey and about 10 other states last spring, were perhaps the most controversial in state history.

Critics, including the state's largest teachers union, questioned the validity of the new tests and bemoaned the time schools spent on test preparation and administration, saying students lost valuable classroom time.

Some parents and students also complained that the tests were overly confusing and questioned whether students were being set up to fail. The backlash led to an "opt-out" movement, and nearly 15 percent of New Jersey high school juniors refused to take the exams.

The New Jersey Education Association, which waged an expensive television advertising campaign against PARCC, said the tests reveal little about student performance.

"Parents and policymakers alike should be very careful about drawing any conclusions from the data released today," NJEA President Wendell Steinhauer said.

Before the results were announced, parent group Save Our Schools NJ had already deemed the data worthless.

"PARCC results mean absolutely nothing," the group said in a statement.

Results for individual schools and students will not be released until next month. Student performance on PARCC will count as a 10 percent factor in some teachers evaluations.

New Jersey students scored lower on PARCC than they did on the prior state exams, the New Jersey Assesment of Skills and Knowledge and the High School Proficiency Assessment.

About 66 percent of third through eighth grade students were rated either proficient or advanced in language arts on the 2013-14 NJ ASK exam, and about 74 percent were proficient or advanced in math.

Nearly 85 percent of students who took HSPA in 2013-14 were proficient or advanced in math and 93 were proficient in English.

But state officials said there is no way to compare PARCC scores to prior tests, which were aligned to different academic standards.

"We have to be very careful when we take one test results and compare it to another," Hespe said.

However, the state did point to other tests that measure career and college readiness and provided similar results for New Jersey students, including the SAT and ACT.

Only 44 percent of students who took the SAT in 2015 met the standards for career and college readiness, Assistant Education Commissioner Bari Erlichson said.

Adam Clark may be reached at adam_clark@njadvancemedia.com. Follow him on twitter at @realAdamClark. Find NJ.com on Facebook.

Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education

BARRY S. FAGIN U.S. Air Force Academy
and
LAURENCE MERKLE Rose-Hulman Institute of Technology

We report the results of a year-long experiment in the use of robots to teach computer science. Our data set compares results from over 800 students on identical tests from both robotics and nonrobotics-based laboratory sessions. We also examine the effectiveness of robots in encouraging students to select computer science or computer engineering as a field of study.

Our results are negative: test scores were lower in the robotics sections than in the nonrobotics ones, nor did the use of robots have any measurable effect on students choice of discipline. We believe the most significant factor that accounts for this is the lack of a simulator for our robotics programming system. Students in robotics sections must run and debug their programs on robots during assigned lab times, and are therefore deprived of both reflective time and the rapid compile-run-debug cycle outside of class that is an important part of the learning process. We discuss this and other issues, and suggest directions for future work.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science

Education *Computer and information science education;* I.2.9 [**Artificial Intelligence**]: Robotics *Commercial robots and applications; Operator interfaces*

General Terms: Experimentation, Languages, Measurement

Additional Key Words and Phrases: Lego Mindstorms, Ada, robots

1. INTRODUCTION

Educators have thought about robots in the classroom for as long as they have thought about robots: their potential as teaching tools and as motivators has long been recognized. For most of our lifetimes, however, economic constraints prohibited extensive deployment of robots in all but the most rarefied environments. Initial attempts to capitalize on robots as teaching tools had to rely on software models [Pattis 1981].

Within the past few years, however, improvements in performance and cost have changed the picture dramatically. Robotic systems, both customized and mass-produced, are now sufficiently affordable, powerful, and reliable to be deployed in the college and even the high school classroom. Interest in the use of robots as educational tools has exploded within the past few years [AAAI 2001; Beer and Chiel 2001; Flowers and Gossett 2002; Harlan 2001; Klassner 2002; Wolz 2001]. We believe this trend will only increase as robots continue to get better and cheaper.

Author s Address: B.S. Fagin is with the U.S. Air Force Academy, USAFA, CO 80840. L. Merkle is with Rose-Hulman Institute of Technology, 5500 Wabash Avenue,Terre Haute, IN 47803

ACM Journal of Educational Resources in Computing, Vol. 2, No. 4, December 2002, Pages 1-18.

But while educators interested in robotics have developed important prototype systems and reported on their deployment in the classroom, very little is known about their effect on learning. The computer science education community has organized panels and workshops on the subject to facilitate the exchange of ideas [AAAI 2001; Congdon 2001], but quantitative studies that assess how robots affect learning are missing from the literature.

We report one such study here. We were among the first institutions to deploy robots in the classroom, so we have considerable interest in this question. The existence of a core course at our institution that all students must take provides a large sample population from which to draw conclusions, and the systematic use of databases for student information and test scores provide us with solid, reliable data for analysis. In the sections that follow, we present the basic parameters of the study, a statistical analysis of the data, and comment on subjective feedback measures. We then present our conclusions and discuss future work.

2. EXPERIMENTAL PARAMETERS

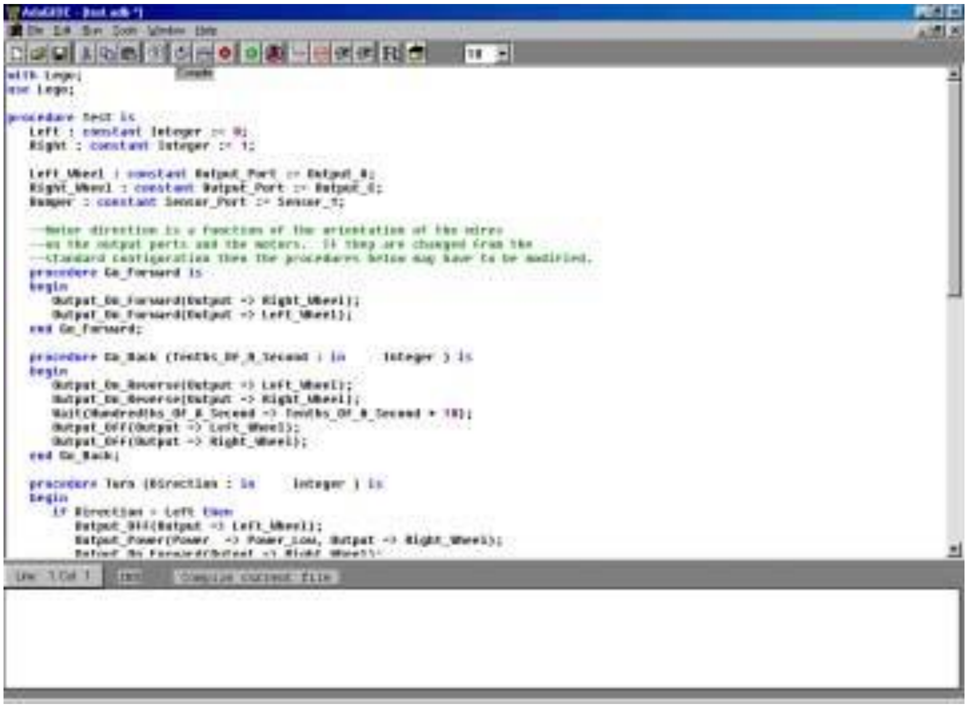
Our study analyzes data from the 2000-2001 academic year offerings of our core computing course, required for all our students and normally taken in the freshman year. This course was taught to 938 students in 48 sections of 15-20 students each. Nine of these sections were designated as robotics sections, where we provided laboratory instruction using Lego Mindstorms fi robots and the Ada/Mindstorms programming environment [Fagin 2000b]. We tracked student performance on all exams, as well as their rank in the course after grades were assigned. Additionally, our students declare a major no later than the middle of their sophomore year, so we now have data on the effectiveness of robots in encouraging the selection of computer science or computer engineering as a field of study.

2.1 The Ada/Mindstorms 2.0 Programming Environment

A screen shot of the Ada/Mindstorms 2.0 programming environment is shown in Figure 1. It has an easy to use GUI, and runs on any Windows PC. It is available for free at http://www.usafa.af.mil/dfcs/adamindstorms.htm.

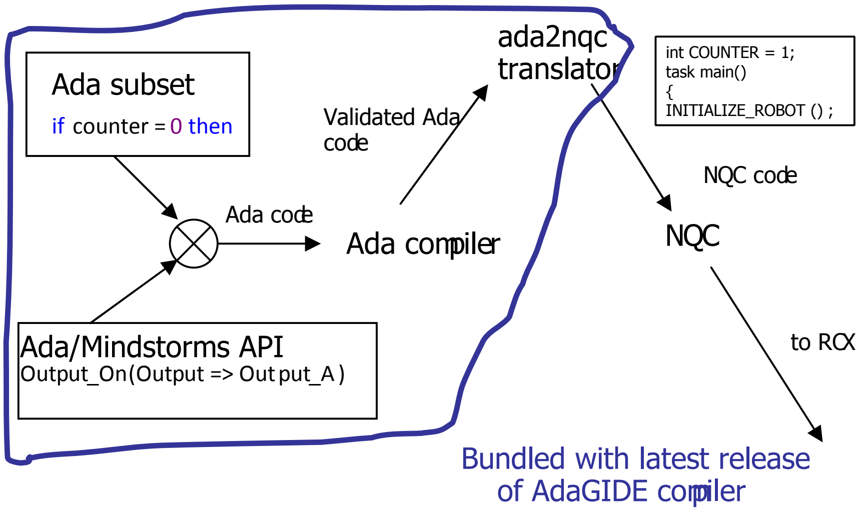
The coding flow of the programming environment is shown in Figure 2. Programs are written in an Ada subset plus an API of Mindstorms-specific function calls, and compiled with the Ada/Mindstorms compiler, which is a fully validated Ada compiler, with additional logic to check that the program uses only those constructs supported by the Ada/Mindstorms subset. These constructs are necessarily much smaller than the full Ada language, due to both Ada s considerable expressive power and the hardware limitations of the Lego Mindstorms platform.

After the user s Ada program has been validated, it is translated into Dave Baum s NQC language [Baum 2002], a C-like language for Mindstorms programming. NQC code is assembled into binary bytecodes and downloaded into the Lego Mindstorms RCX module, the central component of the Mindstorms system. (For more information on Ada/Mindstorms, see Fagin [2000a; 2000b; 2001; 2002].



```
with Lego;
use Lego;

procedure Test is
   Left : constant Integer := 0;
   Right : constant Integer := 1;

   Left_Wheel : constant Output_Port := Output_A;
   Right_Wheel : constant Output_Port := Output_C;
   Bumper : constant Sensor_Port := Sensor_1;

   --Motor direction is a function of the orientation of the wires
   --on the output ports and the motors.  If they are changed from the
   --standard configuration then the procedures below may have to be modified.
   procedure Go_Forward is
   begin
      Output_On_Forward(Output => Right_Wheel);
      Output_On_Forward(Output => Left_Wheel);
   end Go_Forward;

   procedure Go_Back (Tenths_Of_A_Second : in    Integer ) is
   begin
      Output_On_Reverse(Output => Left_Wheel);
      Output_On_Reverse(Output => Right_Wheel);
      Wait(Hundredths_Of_A_Second => Tenths_Of_A_Second * 10);
      Output_Off(Output => Left_Wheel);
      Output_Off(Output => Right_Wheel);
   end Go_Back;

   procedure Turn (Direction : in    Integer ) is
   begin
      if Direction = Left then
         Output_Off(Output => Left_Wheel);
         Output_Power(Power  => Power_Low, Output => Right_Wheel);
         Output_On_Forward(Output => Right_Wheel);
```

Fig. 1. Creating robot programs with Ada/Mindstorms 2.0.



```
int COUNTER = 1;
task main()
{
INITIALIZE_ROBOT () ;
```

Fig. 2. Ada/Mindstorms programming path.

2.2 Tested Computing Concepts and Methods of Assessment

Both the robotics and the nonrobotics sections were taught introductory programming using Ada. Our core computing course contains six laboratory exercises, emphasizing the following concepts:

lab 1: variables, constants, sequential control flow, procedures without parameters

lab 2: terminal I/O (nonrobotics lab), procedures with input parameters lab 3: condition-controlled iteration, selection

lab 4: count-controlled iteration, procedures with output parameters lab 5: arrays

lab 6: file I/O (nonrobotics lab)

These concepts were taught in both the robotics and the nonrobotics sections, using different programming assignments but reinforcing similar ideas. The exceptions were part of lab 2 and lab 6. Since the I/O capabilities of the Mindstorms robot are extremely primitive and files are not supported by the robotics environment, terminal and file I/O were taught using the same conventional laboratory exercises for all students.

Students in the nonrobotics sections were also taught the concept of *packages*, the Ada mechanism for separate compilation. Packages are not currently supported by Ada/Mindstorms, and therefore we decided not to teach this concept in the robotics sections.

The course concludes with a programming project that ties all the concepts together, usually a game, in which students are allowed to work in teams. The game is typically chosen so that a simple strategy is easy to construct, but a good one takes some thinking. On the last day of class, programs compete against each other in a virtual tournament. We attempted to make both the robotics and the nonrobotics sections as similar as possible by assigning team-based projects built around a strategy. In the nonrobotics sections, teams were given a basic programming environment for the game Othello in the fall semester and Mancala in the spring semester, and competed against each other by implementing strategies for making the next move, given a board configuration. In the robotics sections, one of us developed a jousting game similar to Nim, where robots faced each other on a path marked off with foil stripes and moved forward a certain number of spaces, beaming their moves to their opponent using the RCX s infrared port. The robot that took the last available space is considered to have won the joust.

Students in this course could earn up to 1000 points, not counting extra credit, based on the following scale:

labs 1-5	150	5 labs @ 30 points
lab 6	40	
practica	160	30, 50, and 80 points

midterms	250	2 @ 125 points
final project	100	
final exam	250	
other	50	

Programming practica were in-class programming assignments that had to be completed in a specified time period.

Since the labs, practica, and final project were different for the robotics and nonrobotics sections, we did not compare student performance on those exercises. Our analysis is confined to midterm exam scores, final exam scores, and class rank at the end of the semester.

Midterms and the final consisted of three sections: multiple choice, short answer, and programming. For both robotics and nonrobotics classes, the multiple choice and short answer portions of the test were the same. For the programming portions, only slight modifications were made in a few instances to ensure that correct answers did not require concepts that robotics students had not been exposed to (for example, packages). For the vast majority of programming questions, we used identical problems.

3. ANALYSIS: OBJECTIVE MEASURES

This section describes the statistical analysis of our experimental data, along with the results. The statistical techniques used are described in the Appendix.

3.1 Exam Performance

We wanted to measure the effect that a student in a robotics section has on test performance. The analysis considers the effect on both raw exam scores and on residuals after the effect of student GPA is removed via linear regression. This was done to guard against the possibility that unequal distribution of student academic ability might affect the results. The statistical test chosen for analysis is the Kruskal-Wallis H test.

Our results are unequivocally negative. Strictly speaking, the KW test is only used to determine if the scores from the two populations are different. However, *in every case* where a difference was detected, the scores in the robotics sections were worse. All of these differences remained after attempts to compensate for the correletated effects of GPA.

The relevant sample sizes are presented in Table I. Each semester, approximately 5% of the students are excused from the final exam on the basis of their performance during the semester. Those students are obviously excluded from the analysis of performance on the final exam. They are also excluded from the analysis of overall performance on exams, but they are included in the analysis of performance on the graded reviews.

Table I. Sample Sizes

Population	Graded Events

	Fall 2000		Spring 2001	
	Graded Reviews	Final Exam and Course Total	Graded Reviews	Final Exam and Course Total
Robotics Students	53	50	130	125
Non-Robotics Students	444	417	311	292

Table II. Fall 2000 Data

	Raw scores				GPA-adjusted scores			
Graded Event	^{1}	^{2}	H	p	R_1	R_2	H	p
Exam 1	88.1	91.2	.72	.40	0.0	0.0	.25	.62
Exam 2	107.5	112.0	1.96	.16	-0.1	0.0	.64	.42
Final MC	109.4	114.8	3.79	.05	-0.3	0.0	2.93	.09
Final SA	77.4	83.8	6.02	.01	-0.2	0.0	3.73	.05
Final total	186.7	198.5	7.10	.01	-0.3	0.0	6.09	.01
Course total	764.6	802.7	4.14	.04	-0.2	0.0	3.84	.05

Table III. Spring 2001 Data

	Raw scores				GPA-adjusted scores			
Graded Event	^{1}	^{2}	H	p	R_1	R_2	H	p
Exam 1 MC	44.9	45.5	.59	.44	0.0	0.0	.17	.68
Exam 1 SA	53.0	57.3	27.24	0	-0.4	0.2	24.98	0
Exam 1 Total	97.9	102.9	17.82	0	-0.3	0.1	16.74	0
Exam 2 MC	38.7	38.5	.003	.95	0.1	0.0	.33	.57
Exam 2 SA	54.2	57.4	7.25	.01	-0.2	0.1	5.71	.02
Exam 2 Total	92.9	95.9	4.76	.03	-0.1	0.0	3.06	.08
Final MC	124.1	125.8	1.77	.18	-0.1	0.0	.53	.47
Final SA	77.0	80.0	4.45	.04	-0.1	0.1	3.43	.06
Final total	201.1	205.7	4.10	.04	-0.1	0.0	2.1	.15
Course total	846.4	872.5	8.24	0	-0.2	0.1	6.47	.01

Table IV. Complete Academic Year Data

Graded Event	R_1	R_2	H	p

Exam 1 Total	-0.2	0.1	8.87	0
Exam 2 Total	-0.2	0.0	11.67	0
Final MC	-0.1	0.0	4.44	.04
Final SA	-0.1	0.0	5.62	.02
Final total	-0.2	0.0	6.15	.01
Course total	-0.2	0.1	11.88	0

The results of the analysis for the fall 2000 and spring 2001 semesters are summarized in Tables II and III, respectively, while the results of the analysis for the complete academic year are summarized in Table IV. In each case, following the procedure for the Kruskal-Wallis test, student scores were ranked from highest to lowest, the rank values were summed, and the KW test performed to generate an H value. H is then used to calculate a p-value. For the tables that follow:

$_1$ the mean score for Robotics students
$_2$ the mean for other students H the H-value of the
KW test p the p-value of the KW test
R_1, R_2 the residuals from a linear-regression function used to remove the effects of GPA.

In this context, the p-value is the probability that it would be an error to conclude that there is an effect on the exam score in question (i.e., the higher the p-value, the less evidence of a difference). The p-values less than .005 appear as 0 in the tables. Negative values for residuals indicate scores below what a linear function based on GPA would have predicted.

For the fall 2000 semester, raw scores are available for both graded reviews, both the multiple choice and short answer portions of the final exam, and final course standing. As shown by the p-values in Table II, the raw scores of the robotics students are significantly different from those of other students on both portions and the total of the final exam, as well as the overall course standing.

For the spring 2001 semester, raw scores are available for both the multiple choice and the short answer portions of both graded reviews and the final exam, as well as for final course standing. There are significant differences between the robotics and the nonrobotics scores on the short answer portions and the overall scores of all three exams, as well as the final course standing, but not on the multiple choice portions of the exams. With the exception of the overall score on the final exam, the same differences are present in the GPA-adjusted scores.

The raw scores from the fall 2000 semester are not directly comparable to those from the spring 2001 semester. However, as explained in the Appendix, the GPA-adjusted scores from the two semesters are directly comparable. Significant differences exist in all those scores.

3.2 Controlling for Instructors

While space precludes a detailed analysis of data broken down by individual instructors, we did have some instructors who taught both robotics and nonrobotics sections.

One instructor taught sections of both the robotics and nonrobotics versions in the fall 2000 semester. There were significant differences between the raw scores of that instructor s robotics and nonrobotics students on the second graded review and the multiple choice portion of the final exam. Only the former difference remains in the GPA-adjusted scores.

Two instructors taught sections of both the robotics and nonrobotics versions in the spring 2001 semester. For one instructor, there were significant differences between the robotics and nonrobotics raw scores for the short answer portion of the final exam, the total score on the final exam, and the final course standing, but no significant differences

Table V. Computer Science Majors

	Fall 2000	Spring 2001	AY 2000-01
Robotics students declaring CS major	0	2	2
Other students declaring CS major	15	13	28
Total	15	15	30
c_1	5	8	10

Table VI. Computer Engineering Majors

	Fall 2000	Spring 2001	AY 2000-01
Robotics students declaring CE major	1	3	4
Other students declaring CE major	4	6	10
Total	5	9	14
c_1	3	6	6
c_2	< 0	0	0
p-value	0.865	1.000	0.568

Table VII Computer Science and Computer Engineering Majors

	Fall 2000	Spring 2001	AY2000-01
Robotics students declaring CS or CE major	1	5	6
Other students declaring CS or CE major	19	19	38
Total	20	24	44
c_1	6	12	14
c_2	< 0	3	4
p-value	0.699	0.478	0.424

in the GPA-adjusted scores. For the other instructor, there were significant differences in the raw scores on the multiple choice portions of both graded reviews, but the only difference in the GPA-adjusted scores is on that portion of the second graded review. Our conclusion is that controlling for instructor variation reduces the observed negative effects of robots, but does not eliminate it.

3.3 Field of Study

We also attempted to determine the effect that being in a robotics section has on a student s likelihood of declaring a major of either computer science or computer engineering. The statistical test chosen for the analysis of this question is the FisherIrwin test. Like Kruskal-Wallis, Fisher-Irwin is used to determine whether or not two samples come from identical populations, but is considered the appropriate test when data has only two possible values (in this case declaring one of our target majors or not).

The raw data is summarized by time period in Tables V, VI, and VII. Table V shows data for computer science, Table VI for computer engineering, and Table VII for both.

There are no statistically significant results for either the fall or the spring semester. Over the course of the complete academic year, the cadets in the robotics sections were slightly less likely to eventually declare the computer science major.

4. ANALYSIS: SUBJECTIVE MEASURES

In addition to the objective measures described previously, we also looked at subjective measures like student self-assessments of their learning, how students rated the course, and so forth. We obtained quantitative subjective measures through course critique

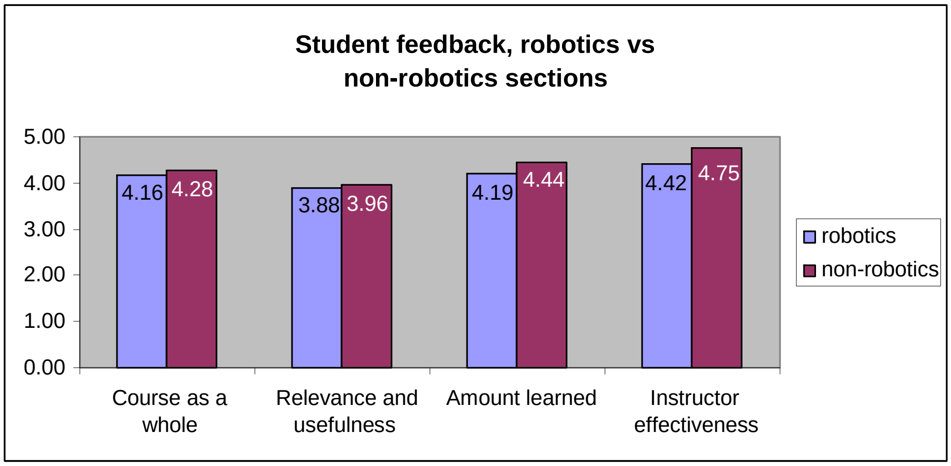


Fig. 3. Student survey results.

Focus Group 1	Focus Group 1	Focus Group 2	Focus Group 2	Focus Group 2

Made you want to learn	Learn something totally new	Visual	Extra instruction always available	Teamwork
No prior Knowledge Needed	Interesting	Problem solving	Problem solving skills	Use of Math
Stimulating	Fun	Practical	Relevant	Plastic
	Challenging	Experimentation	Exposure to new Things	Attention to detail
	Interactive	Magical	More Interesting	Batteries
		Legos		Different
		Funny		
	# of responses			
🟨	4	Fun		
🟥	3	New or Different		
🟦	2	Interesting		
🟧	2	Practical or Relevant		
🟦	2	Problem Solving		
🟪	2	Stimulated Learning		

Fig. 4. Focus group analysis: Advantages of robots (fall semester).

FOCUS GROUP 1			
Unique	Robots	Relevant	Mentally Challenging
Creative	Hands-on	Requires thinking	Hands-on
Interesting	Front page	Fun labs	Creative teacher
Robots	Problem solving	Good instructors	Synergistic
FOCUS GROUP 2			
Organized	Nice Teacher	Learned about computers	Great application to real life computing
Challenging	Interesting fun	Easy to get EI	Logical problem solving skills
Useful	Robots	Worked with applications	Creative
	Real fun	Lots of computers	
FOCUS GROUP 3			

Helped us catch up on our sleep every T-day	Like a leadership reaction courseproblem solving	Problem solving	EI “fun time”
Hands-on	A change from other courses	EI	Instructor xxxxxxx
Legos		Partial credit	Stories
			Excitement

Thread	Color Code	Number of Occurrences
Creative or unique	🟪	5
Interesting or fun	🟥	4
Problem solving	🟩	4
Relevant, useful or applied	🟧	4
Hands-on	🟨	3
Related to EI	🟦	3
Related to instructors	🟩	3
Robots	🟨	3
Thinking or challenging	🟦	3

Note: EI = Extra Instruction, one-on-one tutoring with a student and an instructor GR = Graded Review, a written in-class exam

Fig. 5. Focus group analysis: Advantages of robots (spring semester).

FOCUS GROUP 1	FOCUS GROUP 1	FOCUS GROUP 2	FOCUS GROUP 2	FOCUS GROUP 2
Availability	Hard to work out of class	Wasteful	Had to go too far work on robots	Not very many robots
Time consuming	Takes a lot of time outside of class	Useless	Not enough robots	Books
Amount of work	Irrelevant if Major is Computer Science	Pointless	You get it or you don't, no in between	El far
	Readings not Emphasized	Time consuming	Only spent one Period working on project	Instructors in lab could not help with robots
		Unrealistic		Students to help hard to find
		No focus		Wheels fall off
		Break		
	# of responses			
🟧	4	Workload and Time		
🟩	3	Inconvenient to work with robots		
🟨	3	Hard to get help		
🟪	2	Lack of robots		
🟫	2	Robots Break		
🟦	2	Not relevant		
🟦	2	Books and reading		

Fig. 6. Focus group analysis: Disadvantages of robots (fall semester).

surveys, distributed at the end of every semester as part of the Academy s standard course evaluation protocol.

While quantitative data was important, we were also interested in qualitative feedback. To this end, we selected a few sections of our course, both with and without robotics, for special focus group sessions. Focus group sessions at our institution are led by trained educational assessment specialists, who ask questions and lead brainstorming sessions to determine student perceptions of their classroom experience. To ensure that students respond honestly, instructors do not attend. The sessions are recorded and transcribed, and then analyzed for common themes to pinpoint both strengths and weaknesses.

The results of our student surveys were consistent with the data of the previous section. As we see in Figure 3, on the four most significant survey questions where

FOCUS GROUP 1			
Uppers couldn't help us	Robotics focused GRs	Couldn't work on robots in rooms	Books Blue book-curtain, ADA book
Felt isolated compared to normal CS 110	More lab time	Get rid of double periods	Practicum
Can only use robots in lab	Slacker lab partners	ADA is a poor choice of language	Time
FOCUS GROUP 2			
GRs	Hard to understand	Un-necessary	Would have liked to learn some Adagraph
Slow at times	Time consuming devotion	Boring	Classroom lab instead of separate
Textbooks	Not practical unless going into career	Didn't teach concepts well	
FOCUS GROUP 3			
Lectures may be confusing	Not enough actual lab instruction	Shouldn't be a core course	Error messages
Robots are limited to only one room/lab	Unclear objectives	More time to do labs	Blue screen of death
Robots are limited to motion	Hard to work on in room	More collaboration on labs	Multiple choice on exams

Number of Occurrences	Number of Occurrences	Number of Occurrences
4	4	4
4	4	4
3	3	3
2	2	2
2	2	2

Fig. 7. Focus group analysis: Disadvantages of robots (spring semester students were asked to rate

course content, the average ratings for the nonrobotics sections were higher than those in the robotics classes. The difference ranged from 2.2% for relevance and usefulness to 7.6% for instructor effectiveness.

On the qualitative side, an analysis of the focus group transcripts revealed some important threads. Figures 4 and 5 show a condensed summary of student brainstorming sessions on the perceived strengths of robot-based instruction in the fall and spring semesters. We see that the perceived advantages of robots in the classroom were also felt by students. Words like interesting, fun, challenging, and relevant kept recurring in the discussion.

Unfortunately, brainstorming sessions on the weaknesses of robots revealed just how important the lack of a simulator and the corresponding reduction in reflective problemsolving time was.

Since we (and probably most institutions considering robots) could not afford to purchase kits for every student, and since the inventory control problems were painful to contemplate, we required all robots to remain in the labs. While we tried to hold as many special laboratory sessions as we could, particularly before labs were due, in an environment like the Academy where student free time is at a minimum, this effectively meant that most students could work on their problems only during their assigned lab time. Students were keenly aware of this, and saw it as a big disadvantage.

When we look at Figures 6 and 7 we see that by far the largest concern is that students were not able to work on their programming assignments back in their rooms. Note the frequent occurrence of comments like hard to work on, time consuming, and so on. The more typical complaints that might be expected when working with robots, such as logistical and mechanical issues, were far less significant. This is in some sense encouraging, since the time constraint issues can be solved in software through the addition of a simulator. Solving serious mechanical or logistical issues inherent in the robots themselves would be much more difficult.

5. CONCLUSIONS AND FUTURE WORK

The most desirable goal for introducing robots into the computer science classroom is to improve student learning, so that students can then display evidence of improved learning on tests and problems. Even if no such evidence is found, the case for robots might still be compelling if they improve student retention, attract more people to the discipline, and enhance the classroom experience. Clearly, these goals were not met in our experiment.

Our results show worse results in the robotics vs the nonrobotics sections. At least in our environment, the use of robotics deprives students of the opportunity to work on their code back in their rooms, on their own time, and to practice the write-run-debug feedback loop, which appears to be an important part of the learning process. When we first conceived of this project two years ago, we wanted to deploy the software in the classroom quickly, and therefore made the conscious decision not to include a simulator in the first releases of Ada/Mindstorms. Our hope was that the learning advantages of robots would outweigh the disadvantages of a restricted feedback loop for programming.

Our results do not support this hypothesis.

It may also be argued that, rather than provide a simulator, students should be allowed to take the robots back to their rooms. Although this presents logistical difficulties in our environment, it may not at other institutions. We strongly believe that any efforts to involve robots in teaching CS should permit reflective, out-of-lab, time to work on assignments. In view of our results, we believe that students working with programming robots in traditional universities should be allowed to check out their equipment and take it to their rooms if at all possible. We will continue to explore this option at our institution as well.

Instructor experience may also play a part. Student feedback metrics improve with instructor experience. We collectively had several years teaching the old version of our computing course, with no more than one semester experience teaching the robotics sections. While we were careful to work through all the labs, issue kits to all robotics instructors, and make sure all exercises were carefully worked through and understood before being issued to the students, it is difficult to believe we were completely successful in completely negating the lack of instructor experience with the robots as a factor in student learning. We also have noted previously that, to some extent, controlling for instructor experience appears to mitigate the observed effects of robots on scores.

We also note that our student population, while large, is not representative of the student population as a whole. Most students are not subject to the time pressures of a military academy, nor are they required to take an introductory computer course that must use a high-level programming language. We hope other researchers with different populations will attempt similar studies for comparison with ours.

The next step in this research is to uncouple the effects of robotics from those of reduced access to the programming feedback loop by adding a simulator to Ada/Mindstorms. Due to the enormous size of the design space (Which robots should we simulate? What environment will they operate in?), this presents significant challenges. Based on the results we have seen, our goal is to produce a simulator that runs quickly, is easy to use, and reliably replicates the behavior of simple Mindstorm robots. Hence, students can have a high degree of confidence that once their program works on their computer, it will work in a robot. At the same time, we would like the simulator to function with different robot designs operating in different environments, to maximize the program s usefulness to educators and to enhance the fun factor. This work is currently in progress [Fagin 2003].

APPENDIX: STATISTICAL TECHNIQUES

This section briefly describes the statistical tests used in this research. They are discussed in greater detail in Allen [1990].

A.1 Kruskal-Wallis Test

The Kruskal-Wallis Test determines whether or not two or more sets of data are different enough to justify concluding that the differences should be attributed to something other than random variation. Given k independent samples from k populations, it tests the null hypothesis

H_0: the samples are from identical populations

against the alternative hypothesis

H_1: the populations are not identical.

In this sense it is similar to the well known t-test, but it is applicable whether or not the data are normally distributed.

In brief, the test consists of ranking the data from all of the samples together, computing the total of the ranks R_i for each of the k samples, and then computing the statistic

$$H = \left(\frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} \right) - 3(n+1)$$

H is well approximated by a chi-square distribution for sample sizes of at least 5. Thus, the probability of the differences among the samples occurring randomly is $P[\chi^2 > H]$, where χ^2 is a chi-square random variable with k-1 degrees of freedom. This probability is called the p-value of the test, and if it is less than α then the null hypothesis (that the samples are from identical populations) should be rejected. In that case, the conclusion is that there is a difference between the populations, at the α level of significance.

A.2 Fisher-Irwin Test

The Fisher-Irwin Test, like Kruskal-Wallis, tests whether or not two samples come from identical populations. It differs in that the data in the samples only have two possible values, i.e. the samples are collections of Bernoulli trials. For this reason, the FisherIrwin Test is called a two-sample Bernoulli test. Such tests determine whether or not Bernoulli random variables sampled from two independent populations can be considered to have the same mean.

(We note that it is a slight abuse of a common variety to assume that the students in the Robotics sections and in the non-Robotics sections are drawn from independent populations, because they come from the same student body and are mutually exclusive). Several alternative hypotheses are possible, but as used in this research, the FisherIrwin test tests the null hypothesis

H_0: $p_X = p_Y$

against the alternative hypothesis

H_1: $p_X \neq p_Y$

where p_X and p_Y are the probabilities of success for the two populations.

Procedurally, the first step in the test is to calculate the total number of successes $k = k_X + k_Y$ in the sample data, where k_X and k_Y are the numbers of successes out of the n trials in the first sample and the m trials in the second, respectively. The next step is to determine the critical region, which is set of values of k_X that would be too improbable to be attributed to random variation. For the alternative hypothesis stated above, this includes both values that are improbably large and values that are improbably small. This step is discussed in greater detail below, as is the determination of the p-value of the test. The null hypothesis should be rejected if k_X is in the critical region. It should also be rejected if the p-value is less than the specified level of significance α. Deciding whether to accept or reject the null hypothesis is the last step.

The critical region and the p-value for the test are both based on the hypergeometric distribution. If the random variable X has a hypergeometric distribution with parameters n, k, and N, then

□□□ n □□□ $N-n$

$P[X = k_X]$ = □k_X□□□□□□$Nk-$□□□□ k_X□□□□.

□□□k

This probability is interpreted as the conditional probability of there being exactly k_X successes out of the first n trials, given that there are exactly k successes in all N trials. Specifically, for the alternative hypothesis H_1: $p_X > p_Y$, the critical region is the largest

upper tail of the distribution containing no more than α probability, i.e.

$$C_>(\alpha)=\{i \mid P[X \geq i] \leq \alpha\},$$

and the p-value is $P[X \geq k_X]$. Similarly, for the alternative hypothesis H_1: $p_X < p_Y$, the critical region is the largest lower tail of the distribution containing no more than α probability, i.e.

$$C_<(\alpha)=\{i \mid P[X \leq i] \leq \alpha\}$$

and the p-value is $P[X \leq k_X]$. For the alternative hypothesis used in this research (H_1:

$p_X \neq p_Y$), the critical region is $C_{>}(\alpha/2)\cup C_{<}(\alpha/2)$ and the p-value is twice the smaller of the tail probabilities.

ACKNOWLEDGEMENTS

This work was funded by the USAF Institute for Information Technology Applications, whose support is gratefully acknowledged.

REFERENCES

AAAI 2001 Spring Symposium on Robotics and Education, March 2001, numerous authors.

Allen, A. 1990. *Probability, Statistics, and Queuing Theory with Computer Science Applications*, 2nd ed., Academic Press, San Diego, CA.

Baum, D. 2002. The NQC Web site. Available at http://www.enteract.com/~dbaum/nqc.

Beer, R. and Chiel, H. 1999. Using autonomous robotics to teach science and engineering, *Commun. ACM 42*, 6 (June1999), 85-92.

Congdon, C., Fagin, B., Goldweber, M., Hwang, D., and Klassner, F. 2001. Experiences with robots in the classroom, panel presented at the 32nd SIGCSE Technical Symposium on Computer Science Education.

Fagin, B. 2000a. Using ada-based robotics to teach computer science . In *Proceedings of the 5th International Conference on Innovation and Technology in Computer Science Education* (Helsinki, June 2000), 148-151. Available at http://www.faginfamily.net/barry/Papers/ITICSEWeb/using_ada.htm.

Fagin, B. 2000b. An ada interface for lego mindstorms. *Ada Letters* 20, 3, 20-40. Available at http://www.faginfamily.net/barry/Papers/AdaLetters.htm.

Fagin, B. 2001. Teaching basic computer science concepts with robotics using ada/mindstorms 2.0. In *Proceedings of SIGADA* 01 (Bloomington, MN, Oct. 2001), 73-78. Available at http://www.acm.org/sigada/conf/sigada2001

Fagin, B. 2003. Ada/Mindstorms 3.0: A computational environment for introductory robotics and programming, *IEEE Robotics and Automation*. Special issue on robotics and education, to appear.

Flowers, T. and Gossett, K. 2002. Teaching problem solving, computing, and information technology with robots. Unpublished paper, project description available at http://www.dean.usma.edu/dean/ComputingAtWestPoint/Robots.htm.

Harlan, R. et. al. 2001. The khepera robot and the kRobot class: A platform for introducing robotics in the undergraduate curriculum. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education* (Charlotte, NC, Feb. 2001), 105-109.

Klassner, F. 2002. A case study of LEGO Mindstorms suitability for artificial intelligence and robotics courses at the college level. In *Proceedings of the 33rd SIGCSE Technical Symposium on Computer Science Education* (Feb. 2002), 8-12.

Pattis, P. 1981. *Karel the Robot: A Gentle Introduction to the Art Of Programming*, 2nd ed. John Wiley and Sons,.

Wolz, U. 2001. Teaching design and project management with lego RCX robots. In *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education* (Charlotte, NC, Feb. 2001), 95-99.

Received May 2002; revised February 2003; accepted March 2003