Reprint from:

EVALUATION REVIEW

"Computer Simulation for Program Evaluation"

by William M.K. Trochim and James E. Davis

SAGE PUBLICATIONS, INC.
275 South Beverly Drive
Beverly Hills, California 90212

SAGE PUBLICATIONS LTD
28 Banner Street
London EC1Y 8QE, England

Computer simulations in evaluation research are useful for (1) improving student understanding of basic research principles and analytic techniques; (2) investigating the effects of problems that arise in the implementation of research; and (3) exploring the accuracy and utility of novel analytic techniques applied to problematic data structures. This article describes these uses of microcomputer simulations for the context of human service program evaluation. Simple mathematical models are described for the three most commonly used human service outcome evaluation designs: the pretest-posttest randomized experiment, the pretest-posttest nonequivalent groups design; and the regression-discontinuity design. The models are translated into a single microcomputer program that can be used to conduct the simulations. Examples of the use of this program on an IBM PC microcomputer are provided to illustrate the three uses of the simulations described. The article concludes by arguing that simulations need to utilize experimental design principles when rigorous, definitive results are desired, but that simulations may have great potential value as an exploratory or teaching tool in human service research.

COMPUTER SIMULATION FOR PROGRAM EVALUATION

WILLIAM M.K. TROCHIM
JAMES E. DAVIS
Cornell University

Imagine the teacher faced with the difficulties of explaining evaluation design to a class of students. The teacher has no problem in conveying the importance of defining the evaluation question, understanding the political context of the study, or involving different stakeholder groups in the research process. But when faced with the more formidable "technical" side of the evaluation process—the construction of measures, the choice of a sampling plan, the selection of a research design, and the analysis of the data—the class becomes lost in the complexities of the material. How can the teacher convey the logic behind an analysis of covariance or a pretest-posttest nonequivalent group design in a way that is understandable to the students?

EVALUATION REVIEW, Vol. 10 No. 5, October 1986 609-634

Or, imagine the program evaluator who is in the process of supervising a program evaluation. A number of problems, partially or entirely unanticipated, are beginning to arise. The evaluator is not sure whether all program participants are attending the program or even whether the program is being carried out in a similar way for all participants. Several of the measures for a small subgroup have been lost and the evaluator is having a hard time getting comparison group persons to come back to the agency for posttest measurement. In fact, the evaluator is not even very confident that the program and comparison groups were really comparable to begin with. How can this evaluator examine what the likely effects of some of these problems might be on the final results?

Or, consider the evaluation methodologist who has been exploring a new statistical approach to analyzing a particular program evaluation design. The statistical theory is fairly well developed but requires a number of assumptions about the data—bivariate normal distributions, equivalence of program and comparison group, equal reliability of pretest and posttest measures, and so on. The methodologist is satisfied with the theoretical formulation but is concerned about what might happen if some of the assumptions are not reasonable in practice. How can the methodologist explore the consequences of violating key assumptions and the potential of statistical techniques that attempt to adjust for such problems?

Microcomputer simulation is a tool that can help the teacher, evaluator, and methodologist address these types of questions. In a simulation, the analyst first creates data according to a known model and then examines how well the model can be detected through data analysis. The teacher can show students that measurement, sampling, design, and analysis issues are dependent on the model that is assessed. Students can directly manipulate the simulation model and try things out to see immediately how results change and how analyses are affected. The evaluator can construct models of evaluation problems—making assumptions about the extent or kind of attrition, group nonequivalence, or program implementation—and see whether the results of any data analyses are seriously distorted. The methodologist can systematically violate assumptions of statistical procedures and immediately assess the degree to which the estimates of program effect are biased.

Simulations are better for some purposes than is the analysis of real data. With real data, the analyst never perfectly knows the real-world

processes that caused the particular measured values to occur. In a simulation, the analyst controls all of the factors making up the data and can manipulate these systematically to see directly how specific problems and assumptions affect the analysis. Simulations also have some advantages over abstract theorizing about research issues. They enable the analyst to come into direct contact with the assumptions that are made and to develop a concrete "feel" for their implications on different analysis techniques.

Simulations have been widely used in contemporary social research (Guetzkow, 1962; Bradley, 1977; Heckman, 1981). They have been used in program evaluation contexts, but to a much lesser degree (Mandeville, 1978; Raffeld et al., 1979; Mandell and Blair, 1980). Most of this work has been confined to the more technical literatures in these fields.

Although the simulations described here can certainly be accomplished on mainframe or minicomputers, this article will illustrate their use in microcomputer environments. There are several reasons for preferring microcomputer contexts for simulations. Clearly, the major advantage is the lower costs of running the simulations. Once you have purchased the microcomputer and necessary software there are virtually no additional costs for running as many simulations as are desired. As it is often advantageous to have a large number of runs of any simulation problem, the costs in mainframe computer time can become prohibitive. A second advantage of microcomputers is their portability and accessibility. One can easily move a microcomputer from home to office to classroom or into an agency either to conduct the simulations or to illustrate their use. Students increasingly arrive at colleges and universities with microcomputers that enable them to conduct simulations on their own. Mainframe accessibility, on the other hand, is dependent on dedicated computer facilities or communication devices (such as terminals, modems, or phone lines). There are disadvantages to using microcomputers for computer simulation—slower computing speeds, restrictions on problem size, lower precision in arithmetic operations—but for the program evaluation contexts described here these are often outweighed by the microcomputer advantages of lower cost and greater portability and accessibility.

This article explains and illustrates some basic principles of microcomputer simulation and shows how they may be used to improve the work of teachers, evalutors, and methodologists. The discussion will focus on a specific type of simulation context—the program or outcome evaluation. In program evaluations the goal is to assess the effect or

impact of some program on the participants. Typically, two groups are studied. One group (the program group) receives the program while the other does not (the comparison group). Measurements of both groups are gathered before and after the program. The effect of the program is determined by looking at whether the program group gains more than the comparison group from pretest to posttest. This article will describe how to simulate the three most commonly used program evaluation designs: the randomized experiment, the pretest-posttest nonequivalent group design, and the regression-discontinuity design. Despite the specific context that this work emphasizes, the simulation principles discussed here can be generalized to other research settings including correlational or survey designs.

The remainder of this article will describe the construction of simulation models for the three designs, present a single microcomputer program for simulating all of them, and discuss the applications of such a program for teaching, evaluation implementation, and statistical methodology.

THE SIMULATION MODELS

This work discusses the use of microcomputer simulations for investigating the three most common program evaluation designs used in applied social research. All three designs, in their simplest forms, involve pre- and postprogram measurement of both program and comparison group participants. The three designs differ in the way in which persons are assigned to participate in the program. In the randomized experimental (RE) design, persons are randomly assigned to either the program or comparison group. In the regression-discontinuity (RD) design (Trochim, 1984), all persons who score on one side of a chosen preprogram measure cutoff value are assigned to one group, with the remaining persons being assigned to the other. In the nonequivalent group design (NEGD) (Cook and Campbell, 1979; Reichardt, 1979), persons or intact groups (classes, wards, jails) are "arbitrarily" assigned to either the program or comparison condition. These designs have been used extensively in program evaluations where one is interested in determining whether the program had an effect on one or more outcome measures. The technical literature on these designs

is extensive (see, for instance, Cook and Campbell, 1979; Trochim, 1986) and a discussion of their relative advantages is outside the scope of this article. The general wisdom is that if one is interested in establishing a causal relationship (for example, in internal validity), RE designs are most preferred, the RD design (because of its clear assignment-by-cutoff rule) is next in order of preference, and the NEGD is least preferable.

All three of the program evaluation designs (RE, RD, and NEGD) have a similar structure, which can be described using the notation

$$\begin{array}{ccc} O & X & O \\ O & & O \end{array}$$

where the Os indicate measures and the X indicates that a program is administered. Each line represents a different group; the first line depicts the program participants whereas the second shows the comparison group. The passage of time is indicated by movement from left to right on a line. Thus, the program group is given a preprogram measure (indicated by the first O), is then given the program (X), and afterward is given the postprogram measure (the last O). The vertical similarity in the measurement structure implies that both the pre- and postmeasures are given to both groups at the same time. Model-building considerations will be discussed separately for each design.

THE RE MODEL

We can begin by constructing the preprogram measure for this design. We make the initial assumption that the preprogram measure, X, is the additive function of two components—a true score, t, and a random error factor e_x such that

$$x = t + e_x$$

For each case (or hypothetical person) we randomly generate both t and e_x and add these to produce the pretest. Next, a variable, z, which describes group membership (that is, program or comparison group) is constructed such that

$$\begin{aligned} z &= 1 \text{ if } r \leq 0 \\ &= 0 \text{ otherwise} \end{aligned}$$

where

z is a (0,1) dummy-coded assignment variable
r is a random variable that is distributed ~ N(0, σ^2) and is independent of all other terms

To accomplish this, we simply generate for each case a new random variable, r, which is normally distributed with a mean equal to 0 and standard deviation equal to some value σ. The the case is assigned to program (z = 1) or comparison (z = 0) group according to the above-described rule. Finally, we construct the postprogram measure, y, such that for each case

$$y = t + e_y + (gz)$$

where

y is the postprogram measure
t is the same true score defined above
$e_y \sim N(0,\sigma^2)$ and is unrelated to t, e_x, or r
g is the program effect size
z is group membership as defined

For each case, the postmeasure is an additive composite of the same true ability (t) as for the premeasure, an independent error (e_y) and an effect size (gz). It is important to note that the effect (g) is only added to program group cases because, for comparison group cases z = 0 and the product gz therefore also equals 0.

THE NEGD MODEL

In the nonequivalent group design we assign persons or units to conditions *nonrandomly*. As a result, we expect that the two groups may differ systematically in ability as reflected in the preprogram measures. If, for instance, two classrooms or hospital units are arbitrarily assigned to receive the program or not, it is plausible to assume that the two groups will, on average, differ on both the pre- and postmeasures, even in the absence of the program. In simulations, we can deliberately create

such nonequivalence by adding some constant value to both the pre- and postmeasures for one of the groups. Therefore, in this design, before we can construct the measures, we first need to create the group assignment variable, z, in the same way as for the RE design:

$$z = 1 \text{ if } r \leq 0$$
$$= 0 \text{ otherwise}$$

where r is defined as before. Once this is accomplished, we can create the preprogram measure, x, such that for each case

$$x = t + e_x + (dz)$$

where t is a true score and e_x is a random error factor. Here, d is a constant that is added to the program group (note that it can be either positive or negative depending upon whether one wishes the program group to be "advantaged" or "disadvantaged" relative to the comparison cases) through multiplication with the (0,1) dummy-coded group assignment variable. The postmeasure, y, is constructed for each case, such that

$$y = t + e_y + (dz) + (gz)$$
$$= t + e_y + z(d + g)$$

where

- y is the postprogram measure
- t is the same true score as used to construct x
- $e_y \sim N(0,\sigma^2)$
- d is the constant representing group nonequivalence as used to construct x
- g is the program effect size
- z is the (0, 1) group membership indicator

THE RD MODEL

The model for the RD design can be constructed by beginning with the premeasure, x, such that for each case

$$x = t + e_x$$

where the preprogram measure x is the additive function of a true score, t, and a random error factor, e_x. Next, the group membership variable, z, can be constructed for each case such that

$$z = 1 \text{ if } x \leq (\text{cutoff value})$$
$$= 0 \text{ otherwise}$$

There are two important points to note. First, one must select a cutoff value on the premeasure. Second, the RD design requires that either low or high scorers be assigned to the program group depending on the nature of the evaluation. If the program involves special training in mathematics that should be given to needy students and the premeasure is an indicator of prior math ability (where low scores indicate poor math performance), then all students scoring *below* some premeasure cutoff value would be given the program (as in the previous formula, which would be appropriate for this compensatory situation). However, if the program involves a novel surgical technique that should be piloted only on the most needy cases and the premeasure is an indicator of the severity of illness (where *high* scores indicate the *greatest* need), persons with premeasure scores *above* some value would be assigned to the program and the formula above would need to be adjusted accordingly.

Finally, the post-program measure, y, is constructed for each case such that

$$y = t + e_y + (gz)$$

which is an identical formula to the one used for the RE design, but differs significantly in the definition of z, the group membership indicator.

SUMMARY OF MODEL-BUILDING PROCEDURES

All three designs have the same structure in that for each a preprogram, postprogram, and treatment (dummy variable) measure is created. However, the models presented show that the designs differ considerably in how these three terms are constructed. The major

difference is the assignment variable, z. In the next section the models will be translated into a single computer program that will be used for simulations. A single integrated program is used so that for any problem the three designs can be directly compared.

THE SIMULATION PROGRAM

The models for the three designs can be easily simulated with a single program. This is illustrated with a program written in the *MINITAB* statistical computing system (Ryan et al., 1982)[1] shown in the Appendix. The first section of the program involves the specification of six constraints (K1-K6) that define the parameters for the simulation. By changing the values of these constraints, one can alter the size of the program effect, the degree of nonequivalence (in the NEGD), the reliability of the measures, and the sample size. The random variables that are needed for all three models (t, e_x, e_y, and r) are generated in four "nran" statements. The next few sections on the listing describe the construction of the x, y, and z variables for the three models. Note that in the sample program a premeasure cutoff value of 0 was chosen for the RD design. Table 1 lists the *MINITAB* variables and variable names that correspond to the pretest (x), group assignment (z), and posttest (y) for the three models.

For each model, the program then prints out the group means and standard deviations. Next, bivariate plots are constructed for each model (on output, the letter A indicates a program case; the letter Z indicates a comparison one; a number indicates the number of cases that fall on the same point; and, an asterisk indicates that more than nine cases fall on the same spot). All three designs are analyzed using the same ANCOVA regression model:

$$y_i = b_0 + b_1x_1 + b_2z_i + e_i$$

where

y_i = posttest score for case i
b_0 = constant or intercept parameter
b_1 = linear slope of y on x parameter

TABLE 1
Index of MINITAB Variables and Variable Names for the RE, NEGD, and RD Models

	Variable and Name		
	RE	*NEGD*	*RD*
Pretest (x)	C5 x-RE-RD*	C10 x-NE	C5 x-RE-RD
Group (z)	C6 z-RE-NE	C6 z-RE-NE	C8 z-RD
Posttest (y)	C7 y-RE	C11 y-NE	C9 y-RD

*This is the x variable for both the randomized experiment and regression discontinuity designs.

x_i = pretest score for case i
b_2 = program effect parameter
z_i = group assignment for case i
e_i = residual for case i

In each analysis, the three estimated parameters, b_0, b_1, and b_2, are saved and the key estimate, b_2—the estimate of the program effect—is stored in a new variable. In this way, the results are cumulated over successive runs of the simulation.

The program can be executed interactively by typing each command as presented in the Appendix (note that commands beginning with # are comments and need not be typed). Alternatively, the program can be stored in a standard system file and executed n times using the *MINITAB* command

execute 'filename' n

All examples presented here were run on an IBM PC/XT microcomputer equipped with an 8087 math coprocessor chip.

SIMULATION VARIATIONS AND APPLICATIONS

There are a number of ways in which the simulations described here (and simple variations of the program provided previously) can be

useful in program evaluation contexts. First, they provide a powerful teaching tool (Eamon, 1980; Lehman, 1980). Students of program evaluation can explore the relative advantages of these designs under a wide variety of conditions. In addition, the simulations show the student exactly how an analysis of these designs could be accomplished using real data. Second, the simulations provide a way to examine the possible effects of evaluation implementation problems on estimates of program effect (Mandeville, 1978; Raffeld et al., 1979; Trochim, 1984). Just as NASA explores difficulties in a space shuttle flight using an on-ground shuttle simulator, the data analyst can examine the possible effects of attrition rates, floor or ceiling measurement patterns, and other implementation factors. Finally, simulations make it possible to examine the potential of new data analysis techniques. When bias is detected in traditional analysis and analytic solutions are forthcoming, simulations can be a useful adjunct to statistical theory.

APPLICATIONS FOR TEACHING

To illustrate the utility of the simulation program for teaching, two sets of simulations were run. In the first, the program effect for all models was 5 points, the NEGD program group had a 3-point "advantage" (that is, was nonequivalent on pre- and postmeasures for the NEGD), the premeasure cutoff value was zero for the RD design, the reliability of the measures was equal to .9 (see further on), and there were 100 cases in each of the 50 runs. In the second, all simulation parameters remained the same except that the reliability of the measures was.5, considerably lower than before. The reliability of the measures was set by varying the relative size of the standard deviations of the true and error scores. Reliability is defined as

$$\text{rel} = \frac{\text{var}(t)}{\text{var}(t) + \text{var}(e)}$$

Therefore, if we set K3 in the program (standard deviation of the true scores) equal to 3 and K4 (standard deviation of the error scores) equal to 1, we obtain the reliability

$$\text{rel} = \frac{3^2}{3^2 + 1^2}$$

$$= \frac{9}{10}$$

$$= .9$$

for the first "high reliability" simulations. In the second "low reliability" simulations, we set K3 = 3 and K4 = 3 and thereby obtain

$$\text{rel} = \frac{3^2}{3^2 + 3^2}$$

$$= \frac{9}{18}$$

$$= .5$$

for the reliability of the measures.

The cumulative results for 50 runs for these two simulations are shown in Table 2.

The results illustrate some important methodological principles. First, both the RE and RD designs yield unbiased estimates. In general, we would consider estimates to be unbiased if the average gain does not differ positively or negatively by more than two standard error units from the true gain (that is, a .05 significance level where the gain, g, falls within the interval $b_2 \pm 2SE_{b2}$). For instance, for the RE design, low reliability simulations, the average gain is 4.89 and the standard error is .081. Therefore, the true gain, 5 points, falls within the interval $4.89 \pm 2(.081)$ and the RE design can be considered unbiased for these conditions. Second, the NEGD is shown to yield biased estimates of effect for both low- and high-reliability simulations. This is consistent with the literature on this design (Reichardt, 1979) which maintains that the ANCOVA analysis will yield biased estimates of effect when the pretest is not perfectly measured (that is, there is measurement error on the pretest). Finally, the results show that the designs differ in efficiency. For both the high- and low-reliability simulations the RE and NEGD have similar standard errors of the average gain, whereas the RD design standard errors are considerably larger. This also is consistent with the literature. Goldberger (1972), for instance, demonstrated that, all things

TABLE 2
Simulation Results for the Basic Program
(true gain = 5.0, 50 runs, n = 100 per run)

		b_2	$SE\ (b_2)$	$min\ (b_2)$	$max\ (b_2)$
High reliability (.9)	RE	4.973	.041	4.251	5.578
	NEGD	5.252*	.044	4.448	5.884
	RD	5.032	.066	3.702	5.951
Low reliability (.5)	RE	4.890	.081	3.663	6.401
	NEGD	6.344*	.094	4.521	7.930
	RD	5.030	.180	2.280	7.870

*Significance of coefficient is determined by its value falling outside of the range of 2 standard errors ($b_2 \pm 2SE_{b2}$).

being equal, the RD design requires 2.75 times the number of cases of an RE design in order to have the same relative efficiency.

How can simulations of this type be useful for teaching about program evaluations? First, students can observe the simulation program in progress and get an idea of how a real data analysis might unfold. In addition, the simulation presents the same information in a number of ways. The student can come to a better understanding of the relationships between within-group pretest and posttest means and standard deviations, bivariate plots of pre- and postmeasures that also depict group membership, and the results of the ANCOVA regression analyses. Second, the simulations clearly demonstrate the probabilistic foundations of hypothesis testing in this context. For instance, the results shown in Table 2 illustrate that even with measures that are fairly reliable, one will sometimes obtain estimates of effect that are near the true value (even when the analysis yields biased results on average, as with the NEGD) or estimates that differ considerably from the true value (even when the analysis yields unbiased estimates on average). To demonstrate these notions even more directly, the student can display for each design the histograms of the estimates of effect across a number of simulations runs. Third, the simulations illustrate clearly some of the key assumptions that are made in these designs and allow the student to examine what would happen if these assumptions are violated. For instance, the simulations are based on the assumption that within-group pre-post slopes are linear and that the slopes are equal between groups. The effects of allowing the true models to have treatment interaction

terms or nonlinear relationships can be examined directly with small modifications to the simulation program as Trochim (1984) illustrated for the RD design. Fourth, the simulations demonstrate the importance of reliable measurement. By varying the ratio of true score and error term variances, the student can directly manipulate reliability and show that estimates of effect become less efficient as measures become less reliable. Finally, simulations are an excellent way to illustrate that apparently sensible analytic procedures can yield biased estimates under certain conditions. This is shown most clearly in the simulations reported in Table 2 for the NEGD. Although the apparent similarity between the design structures of the RE and NEGD might suggest that traditional ANCOVA regression models are appropriate, the simulations clearly show this to be false and thereby confirm the statistical literature in this area (Reichardt, 1979).

APPLICATIONS FOR THE STUDY OF DESIGN IMPLEMENTATION

The validity of estimates from the three designs described here depends on how well they are executed or implemented in the field. There are many implementation problems occurring in typical human service program evaluations—attrition problems, data coding errors, floor and ceiling effects on measures, poor program implementation, and so on—that degrade the theoretical quality of these designs (Trochim, 1984). Clearly, there is a need for improved evaluation quality control (Trochim and Visco, 1985), but when implementation problems cannot be contained, it is important for the analyst to examine the potential effects of such problems on estimates of program gains. This application of simulations is analogous to simulation studies that NASA conducts to try to determine the effects of problems in the functioning of the space shuttle or a communications satellite. There, an exact duplicate of the shuttle or satellite is used to try to recreate the problem and explore potential solutions. In a similar way, the program evaluator can attempt to recreate attrition patterns or measurement difficulties to examine their effects on the analysis and discover analytic corrections that may be appropriate. This is illustrated for two implementation problems. In the first, a simple attrition pattern is constructed; the second examines the problem of posttest ceiling effects.

In order to model attrition patterns we need to make assumptions about what causes attrition in the context at hand. Here, we will make a rather simple assumption for purposes of illustration: that persons (or cases) who are low in *in true ability* on pre- and postmeasures are the most likely attrition cases. This might be the case in educational contexts where it may be the lowest ability students who are lacking motivation or are erratic in attendance and therefore are excluded from the data analysis for want of either a pre- or postprogram score. Similarly, in health or mental health contexts it may be the most needy or the most severely ill who contribute most to the attrition rate. We can operationalize this attrition assumption in a somewhat crude way by excluding all cases in the simulation that have true scores lower than some chosen value. In these simulations, the attrition model was accomplished with the addition of the following program statement immediately after the random generation of the true scores:

recode -100-1.5 C1 '*' C1

This command recodes all values between -100 and -1.5 as missing and puts them in column 1 (c1). (It essentially assigns the *MINITAB* missing value code to all cases having a true score lower than -1.5 and these cases are subsequently removed from the analysis.) As in the previous example, all three models were simulated for both low and high reliability measurement. The average estimates of effect, standard errors, and minimum and maximum estimates are shown in Table 3.

The results suggest several lessons. As in the previous simulation, the RE design appears to yield unbiased estimates for both high- and low-reliability conditions. Although the attrition pattern is systematic with respect to true ability (and is therefore correlated with both the pre- and postmeasures) it is random with respect to the assignment variable, r. The NEGD clearly yields biased estimates, and these are even more biased than in the previous nonattrition simulations. The RD design is clearly biased under the high-reliability model and is marginally biased for the low-reliability condition. This suggests that a greater number of simulation runs (or a larger n for each run) might indicate that the RD design yields biased estimates under this attrition model.

The second example of the use of simulations for investigating implementation problems involves the construction of a ceiling effect on the postprogram measures for all three designs. A ceiling effect occurs when a measure is unable to discriminate between the ability levels of

TABLE 3
Simulation Results for the Attrition Model
(true gain = 5.0, 50 runs, n = 100 per run before attrition)

		b_2	$SE\ (b_2)$	$min\ (b_2)$	$max\ (b_2)$
High reliability (.09)	RE	5.074	.052	4.400	5.873
	NEGD	5.608*	.058	4.713	6.610
	RD	5.202*	.074	3.902	6.654
Low reliability (.5)	RE	4.899	.130	2.819	7.057
	NEGD	6.888*	.125	4.636	8.821
	RD	5.310*	.180	2.990	7.970

*See note, Table 2.

persons who do well on the test. When a test is too easy, for instance, many respondents may achieve perfect scores. The scores cannot be considered accurate indicators of their relative ability because if the test were harder, some respondents would outscore others at this upper level. The problem is especially troubling when it occurs on a postprogram measure that is presumed to reflect program-related gains. Instead, potential gains will be masked by the test's inability to allow higher posttest scores.

A simple model for constructing a posttest ceiling effect was constructed in these simulations by forcing all program cases having a 6.5 or greater on the posttest to be given the posttest ceiling value of 6.5 instead.[2] This is easily accomplished by inserting the following three statements immediately before naming the variables in the program:

```
reco 6.5 100 C7  6.5 C7
reco 6.5 100 C9  6.5 C9
reco 6.5 100 C11 6.5 C11
```

In *MINITAB,* the recode command can also be stated as *reco.* The average estimates of effect, standard errors, and minimum and maximum estimates for both the high- and low-reliability conditions are shown in Table 4 for the three designs.

In this example, all three models yield biased estimates of effect for both high- and low-reliability conditions. In all cases but one, the bias is in the direction of *underestimating* the true effect. This is not surprising

TABLE 4
Simulation Results for Posttest Ceiling Effect
(true gain = 5.0, 50 runs, n = 100 per run)

		b_2	$SE\ (b_2)$	$min\ (b_2)$	$max\ (b_2)$
High reliability (.9)	RE	4.242*	.036	3.692	4.959
	NEGD	3.911*	.041	3.465	4.635
	RD	5.218*	.068	4.014	5.987
Low reliability (.5)	RE	3.801*	.084	2.477	5.310
	NEGD	4.267*	.083	3.056	5.686
	RD	4.050*	.150	0.950	6.310

*See note, Table 2.

given that there was a posttest ceiling that prevented larger gains from occurring. In the only exception, the RD design under the high reliability condition, the effect is *overestimated* due to the nature of the regression model that is used. A more detailed consideration of this result is outside the scope of this article and the reader is referred to Trochim (1984) for a more extensive discussion of the RD design and the analytic problems that can lead to this pattern of results.

The attrition and posttest ceiling examples illustrate the use of simulations to examine common research implementation problems. The analyst can directly manipulate the models of the problems in order to approximate their reality more accurately and to examine the performance of a design under more varied situations. Such simulations are useful in that they can alert the analyst to potential bias and even indicate the direction of bias under various assumptions.

APPLICATIONS FOR THE INVESTIGATION OF NEW ANALYSES

One of the most exciting uses of simulation involves the examination of the accuracy and viability of "new" statistical techniques that are designed to address the deficiencies of previous models. There are two reasons why simulations are particularly valuable here. First, the conditions that the new analysis addresses may not be easily amenable to mathematical proof that the analysis will yield unbiased estimates. Second, simulations allow the analyst to examine the performance of the analysis under degraded conditions or conditions that do not

perfectly match the mathematical ideal. Thus, simulations can act as a proving ground for new analyses that supplement and extend what is possible through mathematical argument alone.

This application of simulations can be illustrated well by returning to the initial simulations reported in Table 2, where it was shown that the NEGD yields biased estimates of program effect. This bias is well known in the methodological literature (Reichardt, 1979) and results from unreliability (measurement error) in the preprogram measure under conditions of nonspecifiable group nonequivalence. One suggestion for addressing this problem analytically is to conduct what is usually called a reliability-corrected Analysis of Covariance to adjust for pretest unreliability in the NEGD. The analysis involves correcting the pretest scores *separately* for each group using the following formula:

$$x_{adj} = \bar{x} + r_{xx}(x_i - \bar{x})$$

where

x_{adj} = the adjusted or reliability corrected pretest
$\bar{x}$ = the within-group pretest mean
x_i = pretest score for case i
r_{xx} = an estimate of pretest reliability

The analyst must use an estimate of reliability and there is considerable discussion in the literature (Reichardt, 1979; Campbell and Boruch, 1975) about the assumptions underlying various estimates (for example, test-retest or internal consistency). The reader is referred to this literature for more detailed consideration of this issue. The choice of reliability estimate is simplified in simulations because the analyst knows the true reliability (as discussed earlier). In constructing the microcomputer simulation, the analyst needs to separate the program and comparison group pretest scores when applying the correction formula.[3] This adjusted pretest is then used in place of the unadjusted pretest for the NEGD simulations.

To illustrate the correction, simulations were conducted under the same conditions as for Table 1 but with the reliability-corrected ANCOVA analysis also included. The results are presented in Table 5 for the three designs and the corrected NEGD analysis. The same pattern of results as in Table 1 occurs with the RE and RD designs yielding unbiased estimates (although RD is less efficient) and the

TABLE 5
Simulation Results for Reliability-Corrected ANCOVA Analysis
(true gain = 5.0, 50 runs, n = 100 per run)

		b_2	$SE\ (b_2)$	$min\ (b_2)$	$max\ (b_2)$
High reliability (.9)	RE	5.013	.040	4.496	5.709
	NEGD	5.332*	.047	4.706	5.969
	RD	5.117	.066	4.178*	6.208
	NEGD (reliability corrected)	5.050	.048	4.410	5.632
Low reliability (.5)	RE	5.064	.109	3.387	6.756
	NEGD	6.567*	.114	4.926	8.227
	RD	5.020	.170	2.810	8.570
	NEGD (reliability corrected)	4.970	.150	2.560	7.000

*See note, Table 2.

NEGD evidencing biased estimates for both low- and high-reliability conditions. Here, however, the reliability-corrected NEGD analysis clearly yields unbiased estimates, thus lending support to the idea that this correction procedure is appropriate, at least for the conditions of these simulations.

Simulations have been used to explore and examine the accuracy of a wide range of statistical analyses for program evaluation including models for adjusting for selection biases in NEGD (Trochim and Spiegelman, 1980; Muthen and Joreskog, 1984); for correcting for misassignment with respect to the cutoff in RD designs (Campbell et al., 1979; Trochim, 1984), and for assessing the effects of attrition in evaluations (Trochim, 1982).

DISCUSSION

This article describes several simple simulation models that are appropriate for a few, relatively confined situations, namely, the use of three common research designs for evaluating program effects. Nevertheless, the logic of these simulations is easily extended to other relevant research contexts. For instance, many agencies routinely conduct

sample surveys to identify needs and target populations, assess services that are provided, and compare agency functioning with the performance of other similar agencies or with some standard. One would construct simulation models for survey instruments for the same reasons that they are constructed for evaluation designs—to improve teaching and general understanding, to explore problems in implementing the survey (such as nonresponse patterns), or to examine the probable effect of various analytic strategies. The key to doing this would again rest on the statistical model used to generate hypothetical survey responses. A "true score" measurement model is useful, at least for simple simulations, but may have to be modified. For instance, assume that one question on a survey deals with client satisfaction with a particular service and that the response is a 7-point Likert-type format where 1 = very dissatisfied, 7 = very satisfied, and 4 = neutral. The analyst could make the assumption that for some sample or subsample the true average response is a scale value equal to 5 points (somewhat satisfied), and that the true distribution of responses is normal around this value, with some standard deviation. At some point, the analyst will have to convert this hypothetical underlying continuous true distribution to the 7-point integer response format either by rounding or by generating normally distributed random integers in the first place. Such a variable could then be correlated or cross-tabulated with other generated responses to explore analytic strategies for that survey. Similar extensions of the models discussed here can be made for simulations of routinely collected management information system (MIS) information, for data for correlational studies, or for time-series situations, among others.

Simulations are assumptive in nature and vary in quality to the degree that the reality is correctly modeled. When constructing a simulation, it is important that the analyst seek out empirical evidence to support the assumptions that are made whenever this is feasible. For instance, it should be clear that the simulations described here could be greatly enhanced if we had more specific data on how much and what type of attrition typically occurs, what type of floor or ceiling effects are common, what patterns of misassignment relative to the cutoff value typically arise for the RD design, what the typical test-retest reliabilities (for use in reliability-corrected ANCOVA) might be, and so on. Although some relevant data will be available in the methodological literature, all of these issues are context specific and demand that the analyst know the setting in some detail if the simulations are to be reasonable.

One way to approach the assumptive nature of the simulation task is to recognize that reality conditions or constraints in the models need to be examined systematically across a range of plausible conditions. This implies that multiple analyses under systematically varied conditions that are based upon principles of parametric experimental design are needed in state-of-the-art simulation work. This point is made well by Heiberger et al. (1983: 585):

> The computer has become a source of experimental data for modern statisticians much as the farm field was to the developers of experimental design. However, many "field" experiments have largely ignored fundamental principles of experimental design by failing to identify factors clearly and to control them independently. When some aspects of test problems were varied, others usually changed as well—often in unpredictable ways. Other computer-based experiments have been ad hoc collections of anecdotal results at sample points selected with little or no design.

Heiberger et al. (1983) go on to describe a general model for simulation design that allows the analyst to control systematically a large number of relevant parameters across some multidimensional reality space, including the sample size, number of endogenous and exogenous variables, number of "key points" or condition values, matrix eigenvalues and eigenvectors, intercorrelations, least squares regression coefficients, means, standard errors, and so on.

Although rigorous, experimentally based simulations are essential for definitive analysis of complex problems, they will not always be feasible or even desirable for many program evaluation contexts. Instead, it is important to recognize that simulations are a generally useful tool that can be used to conduct more definitive statistical studies but, more realistically for program evaluation, provide the analyst with the means to explore and probe simple relevant data structures for purposes of improving teaching about research, examing research implementation problems and pilot testing analytic approaches for problematic data.

APPENDIX

MINITAB Program to Simulate Three Program Evaluation Designs

```
# MINITAB Program to simulate a simple pretest-posttest
#   randomized experiment (RE), nonequivalent group (NE) design,
#   and regression-discontinuity (RD) design.
#
# Define simulation parameters
#
let k1=5  # k1 is the gain or program effect
let k2=3  # k2 is the selection bias for the NEGD
let k3=0  # k3 is the mean of the true scores
let k4=3  # k4 is the standard deviation of the true scores let
k5=1  # k5 is the standard deviation of the error terms let
k6=100 # k6 is the number of cases desired
#
# Set MINITAB environment parameters
#
batch
noprint
brief
#
# Generate random variables needed
#
nran k6 k3 k4 c1  # generate true score
nran k6 0 k5 c2   # generate pretest error
nran k6 0 k5 c3   # generate posttest error
nran k6 0 k5 c4   # generate assignment error
#
# Construct pretest for RE and RD
#
let c5=c1+c2 # pretest score
#
# Construct z and y for RE
#
reco -100 0 c4 -1 c6
reco 0 100 c6 0 c6
reco -1 c6 1 c6
let c7=c1 + (k1*c6) + c3
#
# Construct z and y for RD
#
reco -100 0 c5 -1 c8
reco 0 100 c8 0 c8
reco -1 c8 1 c8
let c9=c1 + (k1*c8) + c3
#

Computer Simulations...

# Construct x and y for NEGD
#
let c10=c1 + c2 - (k2*c6)
let c11=c1 + c3 - ((k1-k2)*c6)
#
# Name the variables
#
```

(continued)

APPENDIX Continued

```
name c1='true' c2='x-error' c3='y-error' c4='a-error'
name c5='x-RE-RD' c6='z-RE-NE' c7='y-RE' c8='z-RD'
name c9='y-RD' c10='x-NE' c11='y-NE'
#
# Group statistics for randomized experiment
#
table c6;
stats c5 c7.
#
# Group statistics for nonequivalent group design
#
table c6;
stats c10 c11.
#
# Group statistics for regression-discontinuity design
#
table c8,
stats c5 c11.
#
# Bivariate plot for randomized experiment
#
lplot c7 c5 c6
#
# Bivariate plot for nonequivalent group design
#
lplot c11 c10 c6
#
# Bivariate plot for regression-discontinuity design
#
lplot c9 c5 c8
#
# Regression analysis for randomized experiment
#
regr c7 2 c5 c6 c20 c21 c22
pick 3 3 c22 c23
join c23 c31 c31
#
# Regression analysis for nonequivalent group design
#
regr c11 2 c10 c6 c20 c21 c22
pick 3 3 c22 c23
join c23 c32 c32
#
# Regression analysis for regression-discontinuity design

Computer Simulations...

#
regr c9 2 c5 c8 c20 c21 c22
pick 3 3 c22 c23
join c23 c33 c33
#
# Name results variables and display aggregate results
#
name c31='REresult' c32='NEresult' c33='RDresult'
desc c31 c32 c33
```

NOTES

1. Most commonly available statistical packages could be used. Analogous program listings for SPSS[x] and SAS are available upon request from the first author. The *MINITAB* version is presented here because that language is widely available on micros, minicomputers, and mainframes, is relatively inexpensive, and is easy to learn.

2. In order to make the posttest ceiling conditions similar across the three designs it was necessary to alter the assignment procedure for the RD design so that the program group consisted of cases scoring *above* the cutoff value rather than below it. This is accomplished by replacing the three statements used to create the RD assignment measure, c8, with the following:

```
reco -100 c5 -1 c8
reco 0 100 c8 1 c8
reco -1 c8 0 c8
```

Thus, all cases having a pretest greater than or equal to zero are in the program and all remaining cases are in the comparison group. This variation might arise in practice if the program is given to "advantaged" persons (for example, a scholarship or award) or if the premeasure is an indicator of need where high scores indicate greater need. The ceiling effect of 6.5 is arbitrary, but here it is used for purpose of illustration.

3. Several statements are needed in *MINITAB* to accomplish the correction. Immediately before naming the variables, the following statements should be inserted:

```
choose 0 c6 c10 c11 c40 c41 c42
let c43 = (mean(c41)) + (.9*(c41 - mean(c41)))
choose 1 c6 c10 c11 c44 c45 c46
let c47 = (mean(c45)) + (.9* (c45-mean(c45)))
join c40 c44 c35
join c43 c47 c36
join c42 c46 c37
name c35 = 'NEw-z' c36 = 'NEw-x' c37 = 'NEw-y'
erase c40-c47
```

The first "choose" statement selects the x, y, and z values for all *comparison* cases. The "let" statement computes the reliability correction for the comparison cases using the value of .9 for the estimate of reliability (that is, the high reliability simulations). The next two statements perform the same function for the program cases. The three "join" statements rejoin the program and comparison group values into single columns for x, y, and z. The variables are then named and unneeded work variables are erased.

If desired, the analyst can obtain group statistics and a bivariate plot for this design following the format used for the other designs. To obtain the reliability-corrected ANCOVA regression analysis, insert the following statements immediately after the RD regression analysis:

```
regr c37 2 c36 c35 c20 c21 c22
pick 3 3 c22 c23
join c23 c34 cc34
```

The cumulative results can be named and displayed by removing the last "desc" statement and substituting:

```
name c34 = 'NErelcor'
desc c31-c34
```

REFERENCES

BRADLEY, D. R. (1977) "Monte Carlo simulations and the chi-square test of independence." Behavior Research Methods and Instrumentation 9: 193-201.

CAMPBELL, D. T. and R. F. BORUCH (1975) "Making the case for randomized assignment of treatments by considering the alternatives: six ways in which quasi-experimental evaluations tend to underestimate effects," in C. A. Bennet and A. A. Lumsdaine (eds.) Evaluation and Experience: Some Critical Issues in Assessing Social Programs. New York: Academic Press.

CAMPBELL, D. T., C. S. REICHERT, and W. TROCHIM (1979) "The analysis of fuzzy regression discontinuity design: polit simulations." Unpublished manuscript, Department of Psychology, Northwestern University, Chicago.

COOK, T. D. and D. CAMPBELL (1979) Quasi-experimentation: Design and Analysis Issues for Field Settings. Chicago: Rand-McNally.

EAMON, D. E. (1980) "Labsim:" A data-driven simulation program for instruction in research and statistics." Behavior Research Methods and Instrumentation 12: 160-164.

GOLDBERGER, A. S. (1972) "Selection bias in evaluating treatment effects: some formal illustrations." Discussion paper, Institute for Research on Poverty, University of Wisconsin, Madison.

GUETZKOW, H. (1962) Simulation in Social Science. Englewood Cliffs, NJ: Prentice-Hall.

HECKMAN, J. (1981) "The incidental parameters problem and the initial conditions in estimating a discrete time-discrete data stochastic process," in C. F. Manski and D. McFadden (eds.) Structural Analysis of Discrete Data with Economic Applications. Cambridge, MA: MIT Press.

HEIBERGER, R. M., P. F. VELLEMAN, and A. M. YPELAAR (1983) "Generating test data with independent controllable features for multivariate general linear forms." J. of the Amer. Statistical Assn. 78: 585-595.

LEHMAN, R. S. (1980) "What simulations can do to the statistics and design course." Behavior Research Methods and Instrumentation 12: 157-159.

MANDELL, L.M. and E. L. BLAIR (1980) "Forecasting and evaluating human service system performance through computer simulation," pp. 60-67 in American Statistical Association Proceedings, Washington, DC: American Statistical Association.

MANDEVILLE, G. K. (1978) "An evaluation of Title I and model c1: the special regression model." Amer. Education Research Assn. Proceedings, Washington, DC, April.

MUTHEN, B. and K. G. JORESKOG (1984) "Selectivity problems in quasi-experimental studies," in R. F. Conner et al. (eds.) Evaluation Studies Review Annual, Vol. 9. Beverly Hills, CA: Sage.

RAFFELD, P., D. STAMMAN, and G. POWELL (1979) " A simulation study of the effectiveness of two estimates of regression in the Title I model A procedure." Amer. Educational Research Assn. Proceedings, Washington, DC, April.

REICHARDT, C. (1979) "The design and analysis of the non-equivalent group quasi-experiment." Unpublished doctoral dissertation, Northwestern University, Chicago.

RYAN, T. A., B. L. JOINER, and B. F. RYAN (1982) Minitab Student Handbook. North Scituate, MA: Duxbury Press.

TROCHIM, W. (1982) "Methodologically based discrepancies in compensatory education evaluation." Evaluation Review 6, 3: 443-480.

———(1984) Research Design for Program Evaluation: The Regression Discontinuity Approach. Beverly Hills, CA: Sage.

———(1986) Advances in Quasi-Experimental Design and Analysis. San Francisco: Jossey-Bass.

———and C. H. SPIEGELMAN (1980) "The relative assignment variable approach to selection bias in pretest-posttest group designs," p. 102 in Proceedings of the Social Statistics Section, American Sociological Association, Washington, DC, September.

———and R. VISCO (1985) "Quality control in evaluation," in D. S. Cordray (ed.) Utilizing Prior Research in Evaluation Planning. San Francisco: Jossey-Bass.

William M.K. Trochim is Associate Professor in the Department of Human Service Studies in the College of Human Ecology at Cornell University. He is author of a book on quasi-experimental research design entitled Research Design for Program Evaluation: The Regression-Discontinuity Approach. *He has written on a wide range of topics related to program evaluation, including experimental and quasi-experimental research, statistical analyses for selection bias, program implementation, research quality control, statistical simulation, and conceptualization methods.*

James E. Davis is a doctoral candidate in the Department of Human Service Studies in the College of Human Ecology at Cornell University. His areas of interest are statistical simulation, quantitative policy analysis, and computer-based program evaluation and data collection.