SINGLE-CASE AND SMALL-*n* EXPERIMENTAL DESIGNS

A Practical Guide to Randomization Tests

John B. Todman
Pat Dugard

Single-Case and Small-*n* Experimental Designs

A Practical Guide to Randomization Tests

Single-Case and Small-*n* Experimental Designs

A Practical Guide to Randomization Tests

John B.Todman
Department of Psychology, University of Dundee

Pat Dugard
Independent Consultant

Lawrence Erlbaum Associates
Taylor & Francis Group

New York London

This edition published in the Taylor & Francis e-Library, 2009.

To purchase your own copy of this or any of
Taylor & Francis or Routledge's collection of thousands of eBooks
please go to www.eBookstore.tandf.co.uk.

Cover design by Kathryn Houghtaling Lacey

Library of Congress Cataloging-in-Publication Data
Todman, John.
Single-case and small-n experimental designs: a practical guide
to randomization tests/John Todman, Pat Dugard.
p. cm.
Includes bibliographical references and index.
ISBN 0-8058-3554-7 (cloth: alk. paper)
1. Statistical hypothesis testing. 2. Experimental design.
I. Dugard, Pat. II. Title.
QA277.T63 2001
519.5′6–dc21 00–062294

ISBN 1-4106-0094-7 Master e-book ISBN

Acknowledgments

This book grew out of a paper published in Augmentative and Alternative Communication (Todman & Dugard, 1999) and we would like to express our thanks to the Associate Editor, Jan Bedrosian, who handled the review process. The existence of the book probably owes much to Jan's enthusiastic support. We are also grateful to the various reviewers, from whose comments we have benefitted greatly.

We are indebted to Eugene Edgington in two respects. Without his body of work on randomization tests, there would not have been a paper, let alone a book. Additionally, he was kind enough to read an early version of the paper, and his helpful and encouraging comments have been much appreciated.

Dedication

To generations of students who probably taught us more than we taught them.

Contents

Preface

We are, respectively, an academic psychologist with research interests in assistive communication and a statistician with a particular interest in using statistics to solve problems thrown up by researchers. Each has a long-standing involvement in teaching statistics from the different perspectives and needs of psychology students and others whose main interest is in applications rather than the methodology of statistics. The first author "discovered" randomization tests while searching for valid ways to analyze data from his single-case experiments. The second author already knew about the randomization test approach to making statistical inferences in principle but had not had occasion to use it. She developed an interest in his research problems and we began to work together on designs and associated randomization tests to tackle the issue of making sound causal inferences from single-case data, and the collaboration grew from there. It is the experience of both of us that communication between research psychologists and statisticians is often less than perfect, and we feel fortunate to have had such an enjoyable and (we hope) productive collaboration. We hope we have brought something of the effectiveness of our communication across the abstract statistics-messy research divide to this book. The way we approached the project was, broadly, this: The psychologist author produced a first draft of a chapter, the statistician author added bits and rendered it technically correct, and we then worked jointly at converting the technically correct version into a form likely to be comprehensible to clinical researchers with no more than average statistical skills. Where possible, we have tried to avoid using technically abstruse language and, where we have felt it necessary to use concepts that may be unfamiliar to some, we have tried to explain them in straightforward language. No doubt, the end product will seem unnecessarily technical to some and tediously "stating the obvious" to others. We hope, at least, that most readers will not find our treatment too extreme in either direction.

Our first publication together in the area of single-case research was a paper in *Augmentative and Alternative Communication.* That paper described a number of designs pertinent to assistive communication research, together with Minitab programs (macros) to carry out randomization tests on data from the designs. This book grew out of that paper. In general, the first author is responsible for the designs and examples and the second author is responsible for the programs, although this is undoubtedly an oversimplification of our respective roles. Although Minitab is quite widely used, there are plenty of researchers who do not have access to it. By implementing programs in two other widely used packages (Excel and SPSS) we hope to bring randomization tests into the familiar territory of many more researchers. Initially, we were uncertain about the feasibility of these implementations, and it is true that Excel in particular is relatively inefficient in terms of computing time. On the other hand, almost everyone using Windows has access to Excel and many people who would shy away from an overtly statistical package will have used Excel without fear and trepidation.

First and foremost, this book is supposed to be a practical guide for researchers who are interested in the statistical analysis of data from single-case or very small-*n* experiments. That is not to say it is devoid of theoretical content, but that is secondary. To some extent, we have included theoretical content in an attempt to persuade skeptics that valid statistical tests are available and that there are good reasons why they should not ignore them. In

general, we have not set out to provide a comprehensive review of theoretical issues and we have tried to keep references to a minimum. We hope we have provided enough key references to put readers in touch with a more extensive literature relating to the various issues raised.

Our motivation for the book is our conviction that randomization tests are underused, even though in many cases they provide the most satisfactory means of making statistical inferences about treatment effects in small-scale clinical research. We identify two things holding clinical researchers back from the use of randomization tests. One is the need to modify familiar designs to use the tests, and the other is that tests are not readily available in familiar statistical packages.

In chapter 1, we consider the options for statistical analysis of single-case and small-*n* studies and identify circumstances in which we believe that randomization tests should be considered.

In chapter 2, we consider when statistical analysis of data from single-case and small-*n* studies is appropriate. We also consider the strong tradition of response-guided experimentation that is a feature of visual analysis of single-case data. We note that randomization tests require random assignment procedures to be built into experimental designs and that such randomization procedures are often incompatible with response-guided experimental procedures. We identify this need to change basic designs used in single-case research as one of the obstacles to the acceptance of randomization tests. We argue that, whether or not there is an intention to use statistical analysis, random assignment procedures make for safer causal inferences about the efficacy of treatments than do response-guided procedures. This is an obstacle that needs to be removed for reasons involving principles of good design.

In chapter 3, we take a look at how randomization tests relate to other approaches to statistical inference, particularly with respect to the assumptions that are required. In the light of this discussion we offer some conclusions concerning the suitability of randomization tests now that the necessary computational power is available on computers.

In chapter 4, we use examples to explain in some detail how randomization tests work.

In chapter 5, we describe a range of single-case and small-*n* designs that are appropriate for statistical analysis using randomization tests. Readers who are familiar with the principles underlying randomization tests and already know that they want to use designs that permit the application of such tests may want to go directly to this chapter. For each design, we provide a realistic example of a research question for which the design might be appropriate. Using the example, we show how data are entered in a worksheet within the user's chosen statistical package (Minitab, Excel, or SPSS). We also explain how the data will be analyzed and a statistical inference arrived at. Almost all of the information in this chapter is common to the three statistical packages, with some minor variations that are fully described.

Chapters 6, 7, and 8 present the programs (macros) for running the randomization tests within Minitab, Excel, and SPSS, respectively. The programs, together with the sample data for the examples (from chapter 5), are also provided on CD-ROM. Our second obstacle to the use of randomization tests was availability of the tests in familiar statistical packages. We have attempted to reduce this obstacle by making tests for a range of designs available within three widely used packages. More efficient programs are undoubtedly possible

using lower level languages, but we think this misses the point. For researchers who are not computing wizards, we believe that it is not the time costs that prevent them from using randomization tests. Even for the slowest of our package implementations (Excel), the time costs are trivial in relation to the costs of running an experiment. We argue that providing programs in a familiar computing environment will be more likely to encourage researchers to "have a go" than providing very fast programs in an unfamiliar environment.

In planning our programs, we decided to go for consistency of computational approach, which led us to base all of the randomization tests on random sampling from all possible rearrangements of the data, rather than generating an exhaustive set of arrangements where that would have been feasible. This has the added advantage that users can decide how many samples to specify, permitting a trade-off between sensitivity and time efficiency. Another advantage is that programs based on random sampling can be made more general and therefore require less customization than programs based on an exhaustive generation of data arrangements.

These chapters also contain details of sample runs for all of the examples and explanations of how to set up and run analyses on users' own data. For each example, information is provided on (a) the design specification, (b) the function of each section of the macro, (c) how to change the number of samples of data rearrangements on which the statistical conclusion is based, (d) where to find the results of the analysis, (e) results of three runs on the sample data, including average run times, and (f) an example of how to report the statistical conclusion.

In chapter 9, we give readers some help in finding sources of information about randomization tests and in locating other packages that provide some facilities for running randomization tests. We have not attempted to be comprehensive in our listing of alternative sources, but have confined ourselves to the few with which we have direct experience. Also, we do not provide detailed reviews of packages, but attempt to provide enough information to enable readers to decide whether a package is likely to be of interest to them.

In chapter 10, we consider the issue of whether, and in what circumstances, it may be acceptable to relax the requirement for a random assignment procedure as a condition for carrying out a randomization test. Our conclusions may be controversial, but we believe it is important that a debate on the issue takes place. As in classical statistics, it is not enough just to assert the ideal and ignore the messy reality.

In chapter 11, we take up another thorny issue—power—that, from available simulations, looks a bit like the Achilles' heel of randomization tests, at least when applied to some designs. We are less pessimistic and look forward to further power simulations leading to both a welcome sorting of the wheat from the chaff and pressure for the development of more satisfactory designs.

In chapter 12, we attempt to help researchers who want to use variants of the designs we have presented. One of the attractions of the randomization test approach is that a test can be developed to capitalize on any random assignment procedure, no matter how unusual it may be. The provision of commented examples seems to us a good way to help researchers acquire the skills needed to achieve tailored programs. In this chapter we have tried to provide guidance to those interested in writing programs similar to ours to fit their own particular requirements.

Like others before us, we started out on our exploration of randomization tests full of enthusiasm and a conviction that a vastly superior "newstats" based on computational power was just around the corner. We remain enthusiastic, but our conviction has taken a beating. We now recognize that randomization tests are not going to be a panacea—there are formidable problems remaining, particularly those relating to power and serial dependency. We do, however, remain convinced that randomization tests are a useful addition to the clinical researcher's armory and that, with the computing power now available, they are an approach whose time has come.

Chapter 1
*Data Analysis in Single-Case and Small-*n *Experiments*

VARIETIES OF CLINICAL DESIGN

Research involving a clinical intervention normally is aimed at testing the efficacy of the treatment effect on a dependent variable that is assumed to be a relevant indicator of health or quality of life status. Broadly, such research can be divided into relatively large-scale clinical trials and single-case studies, with small-group studies in a rather vaguely specified intermediate position.

There are two problems that clinical designs are intended to solve, often referred to as internal and external validity. *Internal validity* refers to competing explanations of any observed effects: A good design will remove all threats to internal validity by eliminating explanations other than the different treatments being investigated. For example, if a well-designed experiment shows improved life expectancy with a new cancer treatment, there will be no possible alternative explanation, such as "The patients on the new treatment were all treated by an especially dedicated team that gave emotional support as well, whereas the others attended the usual clinic where treatment was more impersonal." *External validity* refers to the general application of any result found: Most results are of little interest unless they can be applied to a wider population than those taking part in the experiment. External validity may be claimed if those taking part were a random sample from the population to which we want to generalize the results. In practice, when people are the participants in experiments, as well as in many other situations, random sampling is an unrealistic ideal, and external validity is achieved by repeating the experiment in other contexts.

A *true experiment* is one in which it is possible to remove all threats to internal validity. In the simplest clinical trials a new treatment is compared with a control condition, which may be an alternative treatment or a placebo. The treatment or control conditions are randomly allocated to a large group of participants, and appropriate measurements are taken before and after the treatment or placebo is applied. This is known as a *pretest-posttest control group design.* It can be extended to include more than one new treatment or more than one control condition. It is one of rather few designs in which it is possible to eliminate all threats to internal validity. The random allocation of participants to conditions is critical: It is the defining characteristic of a true experiment. Here we are using terminology introduced by Campbell and Stanley (1966) in their influential survey of experimental and quasi-experimental designs. They characterized only three designs that are true experiments in the sense that all threats to internal validity can be eliminated, and the pretest-posttest control group design is one of them.

There are well-established statistical procedures for evaluating the efficacy of treatments in true experiments, where treatments (or placebos) can be randomly allocated to a relatively large number of participants who are representative of a well-defined population. Parametric tests, such as *t*-tests, analyses of variance (ANOVAS) and analyses

of covariance (ANCOVAS) generally provide valid analyses of such designs, provided that some technical assumptions about the data populations are approximately met. For example, the appropriate statistical procedure for testing the treatment effect in a pretest-posttest control group design is the ANCOVA (Dugard & Todman, 1995).

In much clinical research, however, the availability of individuals within well-defined categories is limited, making large-*n* clinical trials impractical. It is no solution to increase the size of a clinical population by defining the category broadly. When this is done, the large functional differences between individuals within the broad category are likely to reduce drastically the power of the usual large-*n* tests to reveal real effects. For example, within the research area of aided communication, people who are unable to speak for a particular reason (e.g., cerebral palsy, stroke, etc.) generally differ widely in terms of associated physical and cognitive impairments. For these and other reasons, such as the requirements of exploratory research or a fine-grained focus on the process of change over time, single-case and small-*n* studies are frequently the methodologies of choice in clinical areas (Franklin, Allison, & Gorman, 1996; Remington, 1990).

For single-case designs, valid inferences about treatment effects generally cannot be made using the parametric statistical procedures typi-cally used for the analysis of clinical trials and other large-*n* designs. Furthermore, although there is no sharp dividing line between small-*n* and large-*n* studies, the smaller the sample size, the more difficult it is to be confident that the parametric assumptions are met (Siegel & Castellan, 1988). Consequently, nonparametric alternatives usually are recommended for the analysis of studies with a relatively small number of participants. Bryman and Cramer (1990) suggested that the critical group size below which nonparametric tests are desirable is about 15. The familiar nonparametric tests based on rankings of scores, such as the Mann-Whitney U and Wilcoxon T alternatives to independent *t* and related *t* parametric tests, are not, however, a complete answer. These ranking tests lack sensitivity to real treatment effects in studies with very small numbers of participants. As with the large-*n* versus small-*n* distinction, there is no clear demarcation between designs with small and very small numbers of participants but, as a rough guide, we have in mind a total number of observations per treatment condition in single figures when we refer to very small-*n* studies.

For some very small-*n* designs, procedures known as *randomization tests* provide valid alternatives with greater sensitivity because they do not discard information in the data by reducing them to ranks. Importantly, randomization tests can also deliver valid statistical analysis of data from a wide range of single-case designs. It is true that randomization tests can be applied to large-*n* designs, but the parametric tests currently used to analyze such designs are reasonably satisfactory and the pressure to adopt new procedures, even if they are superior, is slight. It is with very small-*n* and single-case designs, where valid and sensitive alternatives are hard to come by, that randomization tests really come into their own.

We aim, first, to persuade clinical researchers who use or plan to use single-case or small-*n* designs that randomization tests can be a useful adjunct to graphical analysis techniques. Our second aim is to make a range of randomization tests readily accessible to researchers, particularly those who do not claim any great statistical sophistication. Randomization tests are based on randomization procedures that are built into the design of a study, and we turn now to a consideration of the central role of randomization in hypothesis testing.

RANDOM ALLOCATION OF TREATMENTS TO PARTICIPANTS OR TEST OCCASIONS

As we noted earlier, randomization is a necessary condition for a true experimental design, but we need to be a little more specific about our use of the concept. There are two points to be made. First, random allocation is not the same thing as random sampling and, second, random allocation does not apply exclusively to the allocation of participants to treatment conditions. Each of these points is important for the rationale underlying randomization tests.

Random Sampling Versus Random Allocation

Random sampling from a large, well-defined population or universe is a formal requirement for the usual interpretation of parametric statistics such as the *t* test and ANOVA. It is also often the justification for a claim of generalizability or external validity. However, usually it is difficult or prohibitively expensive even to define and list the population of interest, a prerequisite of random sampling. As an example of the difficulty of definition, consider the population of households. Does a landlord and the student boarder doing his or her own shopping and cooking constitute two households or one? What if the landlord provides the student with an evening meal? How many households are there in student apartments where they all have their own rooms and share some common areas? How can the households of interest be listed? Only if we have a list can we take a random sample, and even then it may be difficult. All the full-time students at a university will be on a list and even have a registration number, but the numbers will not usually form a continuous series, so random sampling will require the allocation of new numbers. This kind of exercise is usually prohibitively costly in time and money, so it is not surprising that samples used in experiments are rarely representative of any wider population.

Edgington (1995) and Manly (1991), among others, made the same point: It is virtually unheard of for experiments with people to meet the random sampling assumption underlying the significance tables that are used to draw inferences about populations following a conventional parametric analysis. It is difficult to conclude other than that random sampling in human experimental research is little more than a convenient fiction. In reality, generalization almost invariably depends on replication and nonstatistical reasoning.

The importance of randomization in human experimentation lies in its contribution to internal validity (control of competing explanations), rather than external validity (generalization). The appropriate model has been termed *urn randomization,* as opposed to sampling from a universe. Each of the conditions is regarded as an urn or container and each participant is placed into one of the urns chosen at random. A test based on the urn randomization approach requires that conditions be randomly assigned to participants. Provided this form of randomization has been incorporated in the design, an appropriate test would require that we take the obtained treatment and control group scores and assign them repeatedly to two urns. Then, an empirical distribution of mean differences arising exclusively from random assignment of this particular set of scores can be used as the criterion against which the obtained mean difference is judged. The empirical distribution is simply a description of the frequency with which repeated random assignments of the treatment

and control group scores to the two urns produce differences of various sizes between the means of scores in the two urns. If urn differences as big as the actual difference between treatment and control group means occur infrequently, we infer that the difference between conditions was likely due to the effect of the treatment. This, in essence, is the randomization test, which is explained in detail with concrete examples in chapter 4.

Participants Versus Exposures

The usual example of random allocation given is the random assignment of treatments to participants (or participants to treatments), but we really are talking about the random assignment of treatments to *exposure opportunities,* the points at which an observation will be made. This applies equally to the "to whom" of exposure (i.e., to different participants) and to the "when" of exposure (i.e., the order in which treatments are applied, whether to different participants or to the same participant). Provided only that some random procedure has been used to assign treatments to participants or to available times, a valid randomization test will be possible. This is the case whether we are dealing with large-*n,* small-*n,* or single-case designs, but the option of using a randomization test has far greater practical implications for very small-*n* and single-case designs.

TESTING HYPOTHESES IN SINGLE-CASE AND SMALL-*n* DESIGNS

The reasons randomization tests are likely to have more impact on hypothesis testing in single-case designs (and, to a lesser extent, small-*n* designs) than in large-*n* designs are that the parametric methods applied in the latter are widely available, are easy to use, and, by and large, lead to valid inferences. There is, consequently, little pressure to change the methods currently used. The same cannot be said for methods generally applied to single-case designs.

Time-Series Analysis Using ARIMA

Various approaches have been proposed for the analysis of single-case designs, and Gorman and Allison (1996) provided a very useful discus-sion of these. Among them, time-series analyses, such as the Box and Jenkins (1976) autoregressive integrated moving average (ARIMA) model, provide a valid set of procedures that can be applied to a range of single-case designs in which observations are made in successive phases (e.g., AB and ABA designs). They do, however, have some serious limitations. For example, they require a large number of observations, far more than are available generally in single-case phase designs (Busk & Marascuilo, 1992; Gorman & Allison, 1996; Kazdin, 1976). Furthermore, the procedure is far from straightforward and unlikely to appeal to researchers who do not want to grapple with statistical complexities. Having said that, for researchers who are prepared to invest a fair amount of effort in unraveling the complexities of ARIMA

modeling, provided they are using time-series designs with large numbers of observations (probably at least 50), this approach has much to recommend it.

Adaptations of Classical ANOVA and Regression Procedures

It seems on the face of it that classical ANOVA and least-squares regression approaches might be applicable to single-case designs. It is well known that parametric statistics require assumptions of normality and homogeneity of variance of population distributions. It is also known that these statistical procedures are robust to violations of the assumptions (Howell, 1997). That means that even quite large departures from the assumptions may result in a low incidence of statistical decision errors; that is, finding a significant effect when the null hypothesis is true or failing to find an effect when the null hypothesis is false. This conclusion has to be modified in some circumstances, for example, when group sizes are relatively small and unequal. However, the general tolerance of violations of assumptions in large-n designs led some researchers (e.g., Gentile, Roden, & Klein, 1972) to suggest that parametric approaches can safely be applied to single-case designs.

This conclusion was mistaken. The problem is that there is an additional assumption necessary for the use of parametric statistics, which is often overlooked because usually it is not an issue in large-*n* designs. This is the requirement that errors (or residuals) are uncorrelated. This means that the deviation of one observation from a treatment mean, for example, must not be influenced by deviations of preceding observations. In a large-*n* design with random allocation of treatments, generally there is no reason to expect residuals of participants within a group to be affected by testing order. The situation is very different for a single-case phase design in which phases are treated as analogous to groups and observations are treated as analogous to participants. In this case, where all observations derive from the same participant, there is every reason to anticipate that the residuals of observations that are close together in time will be more similar than those that are more distant. This serial dependency problem, known as *autocorrelation,* was explained very clearly by Levin, Marascuilo, and Hubert (1978), and was discussed in some detail by Gorman and Allison (1996). They came to the same conclusion, that positive autocorrelation will result in many more significant findings than are justified and that, given the high probability of autocorrelation in single-case phase designs, the onus should be on researchers to demonstrate that autocorrelation does not exist in their data before using classical parametric analyses. The legendary robustness of parametric statistics does not extend, it seems, to violations of the assumption of uncorrelated errors in single-case designs.

The underlying problem with Gentile et al.'s (1972) approach is that there is a mismatch between the randomization procedure (if any) used in the experimental design (e.g., random ordering of phases) and the randomization assumed by the statistical test (e.g., random ordering of observations). This results in a gross overestimate of the degrees of freedom for the error term, which leads in turn to a too-small error variance and an inflated statistic. In pointing this out, Levin et al. (1978) also observed that Gentile et al.'s (1972) approach to the analysis of single-case phase designs is analogous to an incorrect analysis of group studies highlighted by Campbell and Stanley (1966). When treatments are randomized between intact groups (e.g., classrooms), such that all members of a group receive the same treatment, the appropriate unit of analysis is the group (classroom) mean, not the individual

score. Similarly, in a single-case randomized phase design, the appropriate unit of analysis is the phase mean rather than the individual observation. As we shall see, the match between the randomization procedures used in an experimental design and the form of randomization assumed in a statistical test procedure lies at the heart of the randomization test approach.

Nonparametric Tests

Although we accept that there will be occasions when the use of time-series or classical analysis procedures will be appropriate for the analysis of single-case designs, we believe that their practical usefulness is likely to be limited to researchers with a fair degree of statistical experience. Fortunately, there is an alternative, simple to use, nonparametric approach available for dealing with a wide range of designs. There are well-known nonparametric "rank" alternatives to parametric statistics, such as Wilcoxon T, and Mann-Whitney U, for use in small-*n* designs, and in large-*n* designs when parametric assumptions may be seriously violated. These tests can also be applied to sin-gle-case designs that are analogous to group designs. These are generally designs in which the number of administrations of each treatment is fixed, in the same way that participant sample sizes are fixed for group designs (Edgington, 1996). Examples given by Edgington are use of the Mann-Whitney U and Kruskal-Wallis tests for single-case alternating treatment designs and the Wilcoxon T and Friedman's ANOVA for single-case randomized block designs. Illustrative examples of these and other designs are provided in chapter 5.

These standard nonparametric tests are known as *rank tests* because scores, or differences between paired scores, are reduced to rank orders before any further manipulations are carried out. Whenever such tests are used, they could be replaced by a randomization test. In fact, the standard rank tests provide approximations for the exact probabilities obtained with the appropriate randomization tests. They are only approximations because information is discarded when scores are reduced to ranks. Furthermore, the statistical tables for rank tests are based on rearrangements (often referred to as *permutations*) of ranks, with no tied ranks, so they are only approximately valid when there are tied ranks in the data (Edgington, 1995). It is worth noting that standard nonparametric tests are equivalent, when there are no tied ranks, to the appropriate randomization tests carried out on ranks instead of raw data.

RANDOMIZATION TESTS AND THE VALIDITY OF STATISTICAL CONCLUSIONS

As we shall see, randomization tests are based on rearrangements of raw scores. As such, within a particular experiment, they provide a "gold standard" against which the validity of statistical conclusions arrived at using other statistical tests is judged (e.g., Bradley, 1968). This is true for nonparametric and parametric tests alike, even when applied to large-*n* group experiments. For example, consider how the robustness of a parametric test is established when assumptions underlying the test are violated. This is achieved by demonstrating that the test does not lead to too many incorrect decisions, compared with a

randomization test of simulated data in which the null hypothesis is true and assumptions have been violated in a specified way. Notice that we qualified our statement about the gold standard status of randomization tests by restricting it to within a particular experiment. This is necessary because, in the absence of any random sampling requirement for the use of randomization tests, there can be no question of claiming any validity beyond the particular experiment. No inference about any larger population is permissible. In other words, we are talking exclusively about internal as opposed to external validity. This does not seem much of a sacrifice, however, in view of the unreality of the assumption of representative sampling underlying inferences based on classical parametric statistical tests applied to experiments with people.

The Versatility of Randomization Tests

An attraction of the randomization test approach to hypothesis testing is its versatility. Randomization tests can be carried out for many single-case designs for which no standard rank test exists. Indeed, when the number of treatment times for each condition is not fixed, as is the case in most phase designs, there is no analogous multiparticipant design for which a rank test could have been developed (Edgington, 1996). Randomization tests are also versatile in the sense that whenever some form of randomization procedure has been used in the conduct of an experiment, an appropriate randomization test can be devised.

This versatility does have a downside, however. It is too much to expect most researchers, whose only interest in statistics is to get the job done in an approved way with the least possible effort, to construct their own randomization tests from first principles. There are numerous ways in which randomization can be introduced into experimental designs and for each innovation there is a randomization test inviting specification. Our aim is to provide a reasonably wide range of the most common examples of randomization procedures, each with "ready to run" programs to do the appropriate test within each of several widely available packages. In fact, the principles underlying randomization tests are, it seems to us, much more readily understandable than the classical approach to making statistical inferences. In our final chapter, therefore, we attempt to provide some guidance for more statistically adventurous readers who are interested in developing randomization tests to match their own designs.

GUIDELINES FOR STATISTICAL HYPOTHESIS TESTING

To summarize, we suggest the following guidelines for selecting a statistical approach to testing hypotheses:

1. For large-*n* designs in which assumptions are reasonably met, the classical parametric tests provide good asymptotic approximations to the exact probabilities that would be obtained with randomization tests. There seems little reason to abandon the familiar parametric tests.
2. For large-*n* designs in which there is doubt about assumptions being reasonably met, standard nonparametric rank tests pro-vide good asymptotic approximations to the exact

probabilities that would be obtained with randomization tests. It is satisfactory to use these rank tests.

3. For small-*n* designs (less than 15 observations per treatment condition) for which standard nonparametric rank tests are available, it is acceptable to use these tests, although their validity is brought more into question as the number of tied ranks increases. There is no disadvantage to using a randomization test and there may well be a gain in terms of precision or validity (Edgington, 1992).
4. For very small-*n* designs (less than 10 observations per treatment condition), the use of randomization tests is strongly recommended. Our admittedly nonsystematic exploration of data sets suggests that the smaller the number of observations per treatment condition and the larger the number of different treatment conditions, the stronger this recommendation should be.
5. For single-case designs with multiple-participant analogues, classical parametric statistics may be acceptable provided that the absence of autocorrelation can be demonstrated, although this is unlikely to be possible for designs with few observations (Busk & Marascuilo, 1992). If the use of parametric tests cannot be justified, the advice is as for Points 3 and 4, depending on the number of observations per treatment condition.
6. For single-case designs with a large number of observations (e.g., a minimum of 50), but without multiple-participant analogues (e.g., phase designs and multiple baseline designs), ARIMA-type time-series analyses may be worth considering if the researcher is statistically experienced or highly motivated to master the complexities of this approach. The advantage gained is that this approach deals effectively with autocorrelation.
7. For single-case designs without a large number of observations and without multiple-participant analogues, randomization tests are the only valid option. Researchers should, however, at least be aware of uncertainties regarding power and effects of autocorrelation with respect to these designs (see chaps. 3 and 11).

These guidelines are further summarized in Fig. 1.1 in the form of a decision tree, which shows rather clearly that statistical procedures other than randomization tests are only recommended in quite restricted conditions. Before going on (in chap. 3) to consider a range of approaches to statistical inference and (in chap. 4) to provide a detailed rationale for randomization tests and examples of how they work in practice, we take up the issue of the relation between statistical and graphical analysis of single-case designs in chapter 2.

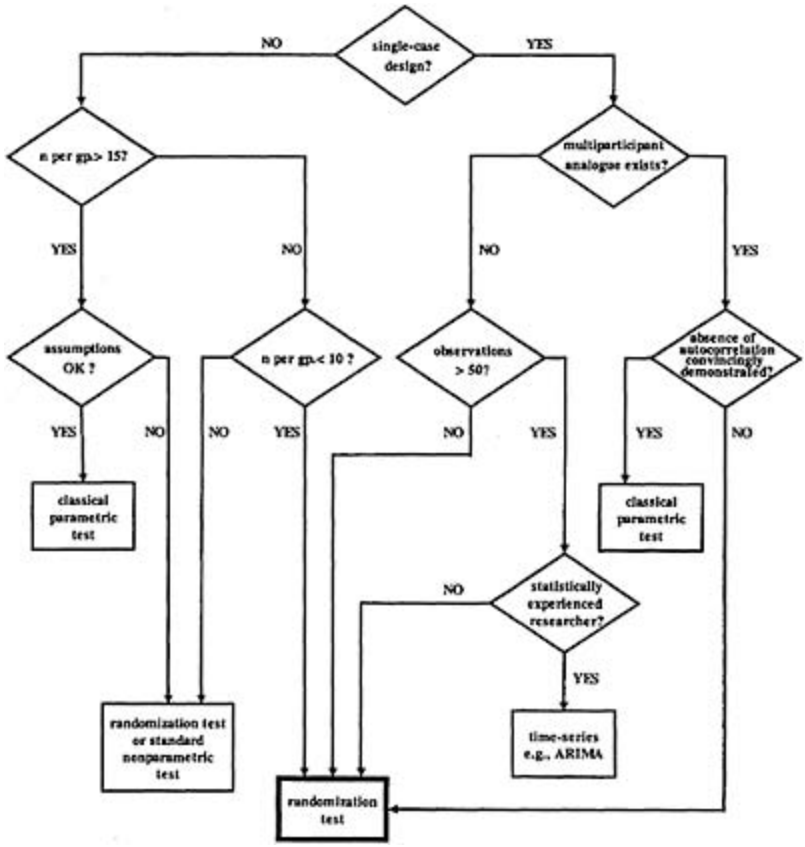

FIG. 1.1. A decision tree for selecting a statistical approach to testing hypotheses.

Chapter 2
Statistical and Visual Analysis

ARGUMENTS AGAINST STATISTICAL ANALYSIS OF SINGLE-CASE DESIGNS

We ended the preceding chapter with some advice about how to select a statistical approach. In a way, we were jumping the gun, in that there has been a good deal of debate about the relevance of statistical analysis to the interpretation of data from single-case studies, and some of the points made could be applied to very small-*n* studies as well. That is the issue we address in this chapter: Should statistics be used at all in single-case designs and, if so, under what circumstances?

The debate has centered around the operant behavioral analysis tradition associated with Skinner (1956) and Sidman (1960) and has extended to a broad range of human clinical intervention studies. The latter vary in their closeness to the operant tradition. Some, reflecting their historical origins within that tradition, are sometimes referred to collectively as *applied behavior analysis,* with a subset that remains particularly close to the reinforcement emphasis of the operant tradition being grouped under the label *behavior modification*.

In the case of the operant tradition, the arguments are in large part directed against traditional group research in general, where descriptive statistics are routinely used to summarize data across participants (Kazdin, 1973; Sidman, 1960). As our principal concern is with single-case designs, it is unnecessary to detail here the arguments about averaging over participants. However, the operant case against the use of inferential statistics applies equally (even especially) to single-case research and we summarize the debate about the use of inferential statistics in single-case designs. In doing this, we attempt to justify our conclusion that there are at least some single-case research situations in which statistical analysis is a desirable adjunct to visual analysis. In addition, we argue that there is a strong case for building randomization procedures into many single-case designs, regardless of whether the data will be analyzed by visual or statistical methods or both. This is particularly relevant to our concern with statistical analysis, however, because, once a randomization procedure has been incorporated in a design, it is a rather straightforward step to make a randomization test available for analysis of the data.

THE OPERANT VIEWPOINT

There are several strands to the operant preference for eschewing statistical analysis in single-case experiments. There are (admittedly highly interwoven) strands concerned with issues of control, response-guided experimentation, and replication.

Control

If an experimenter is able to use an experimental variable (e.g., contingent timing of food pellet delivery for a rat or verbal praise for a person) to gain complete control over a behavioral variable (e.g., rate of bar pressing for a rat or rate of computer key pressing for a person), statistics are not required to support the inference of a causal connection between the variables. Visual inspection of a raw record of changes in response rate will, provided the response record is extensive, be sufficient to show that the rate changes systematically as a function of the contingent timing of the experimental variable. By *contingent timing* we mean timing that is dependent in some way on the behavior of the subject (e.g., delivery of food or praise follows every bar or key press, every fifth press, or every fifth press on average). Skinner (1966) argued that it is control in this sense that should be the goal of the experimental analysis of behavior, and any emphasis on statistical analysis of data simply distracts the researcher from the crucial task of establishing control over variables.

The notion of control is also central in the standard theory of experimental design described by Campbell and Stanley (1966). The internal validity of an experiment depends on the ability of the researcher to exercise control over extraneous (nuisance) variables that offer alternative explanations of the putative causal relation between an independent (manipulated) variable and a behavioral (dependent) variable. If the re-searcher could exercise complete control, then, as in the idealized operant paradigm, no statistical analysis would be required to establish the causal relation. In practice, the control of variables becomes increasingly problematic as we move from rats to humans and as we strive for ecological validity; that is, interventions that yield effects that are not confined to highly constrained laboratory conditions but are robust in real-world environments (Kazdin, 1975). In the absence of extremely high levels of control of nuisance variables, the randomization of treatments is, as we have seen, the means by which the internal validity of an experiment can be ensured.

The internal validity of an experiment depends on there being no nuisance variables that exert a biasing (or confounding) effect. These variables are *systematic nuisance variables,* such as the dedicated team effect favoring the new cancer treatment group in the example in chapter 1. These maybe distinguished from *random nuisance variables*. The emotional state of patients when they visit the clinic is a nuisance variable: something that may affect the outcome regardless of which treatment is applied. It will, however, be a random nuisance variable provided that treatments and attendance times are allocated to patients in a truly random way. Then, even though some will be in a better emotional state than others when they attend the clinic, this nuisance variable is equally likely to favor either group. There are other scenarios, however, in which the effect of differences in emotional states would not be random. For example, if some nonrandom method of allocating treatments to cancer patients had been used, such as assigning the first 50 to attend the clinic to the new treatment group and the next 50 to the old treatment group, it is possible that patients with generally better emotional states would be less inclined to miss appointments and therefore more likely to be among the early attendees assigned to the new treatment group. General emotional state would then be a systematic nuisance variable, which would compromise the internal validity of the study by offering an alternative explanation of any greater improvement found for the new treatment group. Statistical tests can then do nothing to rescue the internal validity of the study. The only remedy is to build random allocation procedures

into the study design at the outset. What random allocation of treatments to participants or times achieves is conversion of potentially systematic nuisance variables (like emotional state) into random nuisance variables.

When potentially systematic nuisance variables have been converted into random variables by the procedure of random allocation, they remain a nuisance, but not because they compromise the internal validity of the study. Rather, they are a nuisance because they represent random variability, for example, between participants in group studies. It is this random variation that makes it hard to see the effects of the independent variable unequivocally. Random effects may happen (by chance) to favor one group or the other. If more patients with good emotional states happened to be randomly assigned to the new treatment group, resulting in a greater average improvement by that group compared with the old treatment group, their greater improvement might be mistaken for the superior effect of the new treatment. This is where statistical analysis comes in. The statistical analysis, in this case, would tell us how improbable a difference between groups as big as that actually obtained would be if only random nuisance variables were responsible. If the probability is less than some agreed criterion (typically 0.05 or 0.01), we conclude, with a clearly specified level of confidence ($p< 0.05$ or $p<0.01$), that the difference was unlikely to have been caused by uncontrolled random variables. That leaves us free to infer (provided the experiment is internally valid) that the difference was probably caused by the independent variable (the superiority of the new treatment in our example).

All that has been said about control of both systematic and random nuisance variables in group studies is equally true for single-case studies. Why should it not be? Why should researchers using single-case experimental designs have a special dispensation to ignore the need for random assignment to ensure internal validity? As Edgington (1984) said, "Randomization is fundamental to inferences concerning treatment effects, whether such inferences are based on graphs or on statistical tests" (p. 87). He went on to suggest that the widespread failure to employ randomization in single-case experiments may be because the function of randomization in experimentation is not generally understood. It is for this reason that we have provided a fairly detailed treatment of the issue.

We cannot emphasize too strongly that the primary reason for random assignment of treatments in experiments is to secure internal validity. This is entirely separate from the question of whether to use statistics, except insofar as randomization makes valid statistical inferences a possibility. The more control the experimenter is able to achieve over random nuisance variables (e.g., by constraining the experimental environment), the less reason there will be to use statistics. Again, this is true both for group experiments and single-case experiments. Researchers who use large group studies generally accept that they lack sufficient control of random variables (not least, random variations among participants) to make inferential statistics unnecessary. Operant researchers, on the other hand, tend to claim that they do achieve sufficient control over random nuisance variables to make statistical analysis unnecessary (e.g., Skinner, 1966). As noted previously, this claim becomes harder to sustain as we move along a continuum from animals in highly constrained environments to humans in close-to-natural environments. However, clinical researchers then justify a disinclination to use inferential statistics on the grounds that they are interested only in large (clinically significant) ef-fects, which are immediately apparent by visual inspection of graphed data even though there is much random variability in the data. This argument is examined a little later.

Another issue that must be addressed is the conflict between the importance of randomization in experiments from which it is intended to draw causal inferences and the operant emphasis on response-guided experimentation, which is shared by many clinical researchers who are not necessarily committed to operant principles.

Response-Guided Experimentation

In operant research, there is a strong emphasis on responsive experimental procedures, which is strongly associated with a preference for graphical methods of data inspection and analysis. Typically, the participant's behavior determines the sequence and pacing of procedures used in an investigation. Researchers make decisions about what to do next and, particularly, about the timing of events under their control, based on behavioral data accumulated (usually in graphical form) as the experiment proceeds. In part, this step-by-step responsive contact with the data is justified in terms of the requirements of *shaping,* which provides the underlying rationale for some designs, such as changing criterion designs (Kazdin, 1980). The justification is also partly on grounds of serendipity, the belief that scientific advances do not always arise from intentional search and that "accidental" discoveries are most likely to occur when researchers show flexibility in following where the data lead. It would certainly be perverse to suggest that researchers should ignore unexpected events, and single-case researchers collectively set a good example by adhering to the principle of staying close to the data more than most between-group researchers.

Responsive experimental procedures are ubiquitous in applied behavioral analysis whenever phase designs (e.g., AB, ABA, etc.) are used, and we focus our discussion on this very popular class of designs. The decision of when to switch from one phase to another, especially from a baseline phase to a treatment phase, is based on the cumulative pattern of behavioral responses in the preceding (e.g., baseline) phase. Widely accepted practice is to continue with baseline observations until the baseline has stabilized, that is, it is reasonably smooth and shows no further evidence of a trend, or at least no trend in the direction predicted for the treatment phase (e.g., Edgington, 1984; Kazdin, 1982). It is frequently claimed (e.g., Shaughnessy & Zechmeister, 1994) that this is essential for interpretation of responses during treatment interventions following the baseline phase.

It is true that if baseline responses are highly variable and particularly if they already show a trend similar to that predicted for the treatment phase it will be difficult to argue that the treatment was effective. It is no solution, however, to resort to response-guided timing of the intervention. In fact, when the timing decision is determined by a rule based on the pattern of baseline responses, any claim to internal validity goes out the window. Why should this be so and what are the alternatives?

Internal validity cannot be established when the timing of the intervention is response-guided because there is no way of telling what the baseline would have looked like if it had been allowed to continue for a few more observations. We know that it probably would have appeared less stable if we had stopped it a few observations earlier, otherwise we would have stopped it then. If the treatment had no effect, it might also have looked less stable if we had waited for a few more baseline observations before introducing the intervention. After all, we stopped when things seemed to be going particularly well. Perhaps that was just a lucky sequence of trials. One of the difficulties is that intuition is often misleading

when we are dealing with probabilities, and in these kinds of experiments we are unlikely to have much idea of the probability of a run of similar responses being followed by a run of increasing or decreasing ones, all of them being nothing more than random variation. In fact, if the responses form an entirely random sequence, then a run of similar values is quite likely to be followed by a run of values that look like a trend up or down. This is the dilemma that always confronts the researcher when there is no randomization of the assignment of treatments to participants or observation occasions. As we have seen, single-case designs are not immune to this problem.

A possible alternative for AB designs would be one in which the total number of available sessions was decided beforehand, along with the minimum number required in the baseline and intervention phases. There would then be random selection of a session in which to begin the intervention from those available. For example, if there were to be a total of 15 sessions, with at least 4 sessions in both baseline and treatment phases, the session in which the intervention is to be introduced would be randomly selected from among Sessions 5, 6, 7, 8, 9, 10, 11, and 12. To overcome the objection that the researcher may not know in advance how many baseline sessions are required to achieve a reasonable level of stability, a simple modification of the randomized intervention design is possible: Run prebaseline sessions until a stability criterion is reached, then begin the baseline sessions followed by the treatment sessions, with the latter commencing at the randomly determined intervention session. This increases the total number of sessions required, but that is a small price for the possibility of an internally valid experiment.

If the usual response-guided procedure is used, it is predictable that Type I errors (i.e., the probability of inferring that the treatment is effective when in fact it is not) will increase. This will be true regardless of whether inferences are based on visual or statistical analysis. With re-gard to visual analysis, Busk and Marascuilo (1992) concluded that visual inspection, as a methodology, "has been shown to have serious deficiencies in replicability" (p. 161). Furthermore, Franklin, Gorman, Beasley, and Allison (1996) concluded from a review of studies "evaluating the performance of visual analysis" (p. 139), that the increase in the percentage of false positive results is likely to be substantial. Just how substantial will certainly depend on characteristics of the data, such as the extent of autocorrelation (the effect of a response in one session on the following responses) and learning or practice effects that occur irrespective of the treatment intervention. This is, nonetheless, a very different picture than that painted by Parsonson and Baer (1978) with their claim that visual analysis is more insensitive, and therefore less prone to Type I errors, than statistical analysis. If single-case researchers use a randomization procedure along the lines just suggested, the bias will be removed whether they analyze their data visually, statistically, or both. Although there is a strong tendency for them to go together, this particular bias results from the use of response-guided intervention rather than from visual analysis as such. We recently demonstrated just how powerful the response-guidance bias can be with an illustrative example (Todman & Dugard, 1999) that we repeat here.

An Illustration of Response-Guidance Bias.

We use a simulated example of a conventional AB design such as might be used to evaluate the effectiveness of an intervention (e.g., assertiveness training) on a behavioral measure

(e.g., frequency of contributions to group discussions by a shy student). Let us assume that in such an AB experiment, baseline sessions were continued until a stability criterion of five consecutive nearly equal scores was obtained, after which assertiveness training was given and posttreatment sessions followed. Now look at Fig. 2.1, which shows data that might have been obtained, with trend lines superimposed. The discontinuity of the data path, together with a vertical line at the intervention point, clearly separates baseline and postintervention (treatment) phases, as recommended by Parsonson and Baer (1978) in their defense of graphical analysis.

It seems fairly compelling to infer from visual analysis of Fig. 2.1 that the treatment was effective. However, because these are simulated data, we know from the way they were generated that there was no systematic discontinuity at the point of intervention in Session 8. In fact, the way the data points were generated assumes that the only systematic effect was a general upward trend over sessions, with random variations of individual points around the trend line. To be precise, the trend line was defined by the equation $y=5+0.2x$.

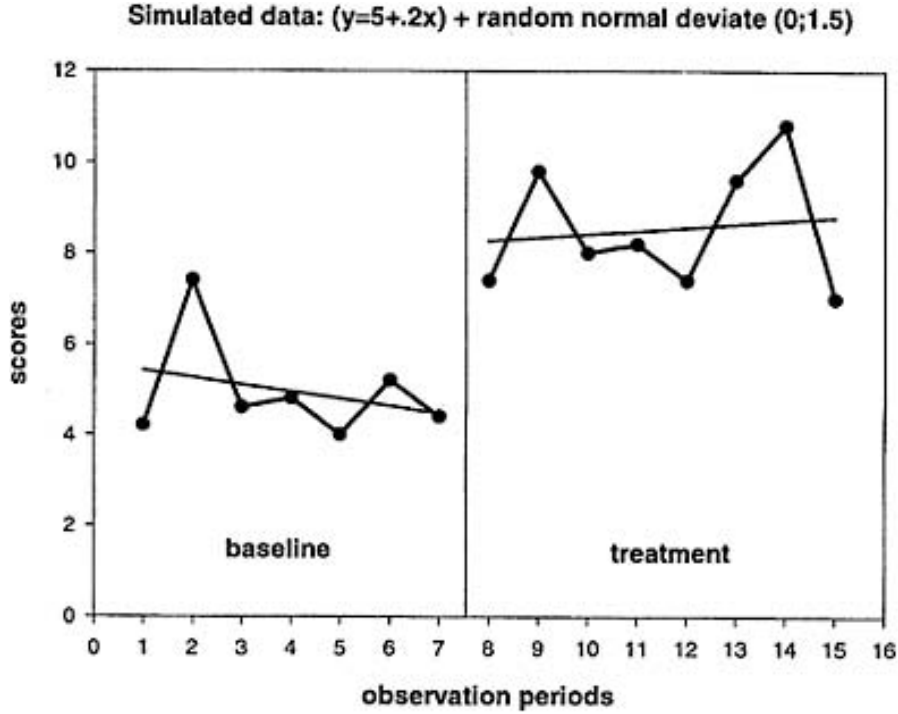

FIG. 2.1. An apparent treatment effect in simulated data for an AB design in which the null hypotheses is true, but a trend over trials exists: Treatment is shown to commence when a baseline stability criterion is reached in observation Period 8.

This means that it starts at a score of 5 on the vertical axis and increases by 0.2 in each session. Each point on the trend line was allowed to vary randomly up or down by an amount related to the frequency of different values along a normal (bell-shaped) curve with a mean of 0 and a standard deviation of 1.5 (i.e., a random normal deviate: 0; 1.5). This is the way numerous small, uncontrolled (chance) effects may be assumed to influence scores, with mostly small positive or negative shifts and progressively larger shifts being more and more rare. The data were therefore completely consistent with existence of a steady practice effect with no additional effect of assertiveness training, because that was the method by which the data were generated. Selection of a stopping point for baseline sessions at which the upward trend was least apparent (i.e., the baseline appeared most stable) made it more likely that a change in trend from baseline to treatment phases would be "seen" even though the treatment had no effect.

To understand why this should be so, we need to look at what happens when the alternative, random procedure for selecting an intervention point is used. Assume that we require at least four sessions in each of the baseline and treatment phases. If, in randomly selecting an intervention point from the available sessions (5–12) we happened to select Session 8 (as in Fig. 2.1), this would produce easily the largest difference between treatment and baseline means (and slopes). If any other intervention point had been randomly selected, we would have been less likely to be misled into inferring that the treatment was effective. This is illustrated in Fig. 2.2, which displays exactly the same data as Fig. 2.1, but with the intervention point occurring two sessions later. Use of a response-guided procedure to select an intervention point gives us our best chance of finding a spurious effect, whether we use statistical or visual analysis.

It is true, of course, that the data in Figs. 2.1 and 2.2 were carefully selected from a number of data sets, generated using the same equation and random normal deviate, to provide a clear illustration of the argument for random determination of the intervention point. Not all such data sets were equally compelling, but it should not be assumed that misleading consequences of response-guided intervention are likely confined to carefully selected data sets. A small experiment was carried out to ascertain the effect of response-guided versus random determination of the intervention point on frequency of incorrect inferences, using the same equation and random normal deviate to generate data sets (Todman & Dugard, 1999).

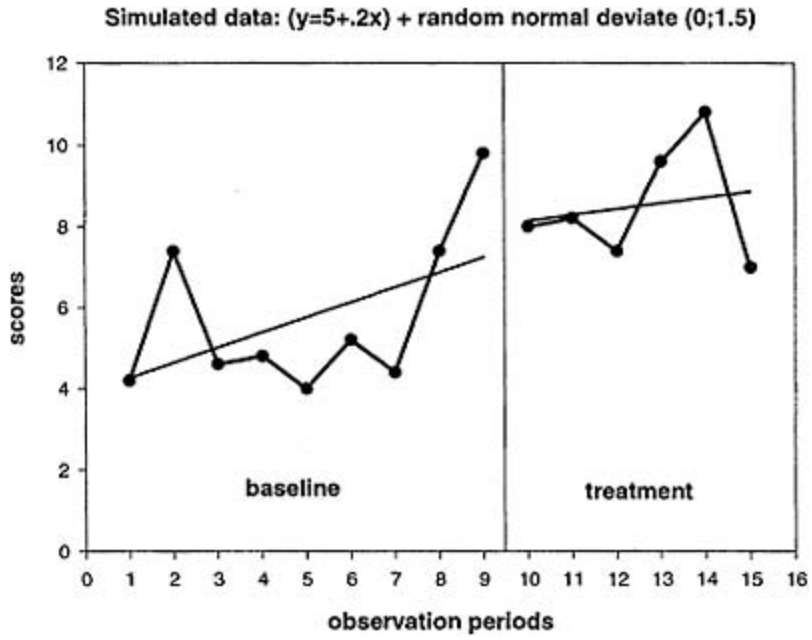

FIG. 2.2. The absence of an apparent treatment effect in the same simulated data as in Fig. 2.1 when treatment is shown to commence in observation Period 10.

An Experimental Test of Response-Guided Bias.

Eighty graphs were generated using the same equation and random deviate as in the example. In 50 of these graphs it was possible to identify several fairly flat points in a row that could be used to designate an intervention point. Two copies of each of the 50 graphs were prepared and a vertical line was drawn on each to indicate the intervention point. On one of each pair, the intervention point followed immediately after the flat points. On the other, the intervention point was randomly assigned to one of the sessions between Session 5 and Session 12, inclusive.

Twelve psychology students, moderately familiar with interpretation of graphical data, were each given the pile of 100 graphs in a different random order and asked to sort them into two piles: those showing an effect of treatment and those not showing an effect. The mean numbers of graphs sorted into the effective treatment pile were 33.0 (*SD*= 6.0) for those with the intervention following the flat points and 11.4 (*SD*=8.6) for those with randomly assigned intervention points. This difference was highly significant in a two-tailed related *t* test, $t\ (11) = 9.84$, $p<0.001$), and the method of assignment of the intervention point accounted for 90% of the variance. Although the severity of the problem may vary with different baseline trends, degrees of random variability and autocorrelation in the data, and experience of judges, our results are consistent with those reported by Franklin, Gorman, Beasley, and Allison (1996). It is difficult to resist the conclusion that

response-guided assignment of treatment intervention points is indeed likely (as consideration of the logic of the procedure predicts) to result in more false positive inferences. Edgington (1984) and Franklin, Gorman, et al. (1996) concurred with the conclusion we draw from our experiment.

Statistical Consequences of Response-Guided Experimentation.

Apart from the internal validity problem occasioned by response-guided procedures in phase designs, which applies regardless of whether visual or statistical analysis is applied to the data, they have the added disadvantage of ruling out the possibility of valid statistical analysis in many cases. We saw in chapter 1 that some form of random procedure in the assignment of treatments to participants or to available observation times is a prerequisite for the use of randomization tests. Because these are the only statistical tests that are both valid and practical for some single-case designs (e.g., phase designs without an unusually large number of observations), use of nonrandom, response-guided procedures in these designs effectively makes them nonamenable to statistical analysis, even if such an analysis is desired.

Proponents of applied behavioral analysis may argue that the nonavailability of valid statistics is of little consequence, visual analysis always being the method of choice because it is less sensitive than statis-tical analysis. This is considered desirable because it means that large (clinically significant) effects will be picked up and statistically significant, but clinically unimportant, effects will be excluded. It is worth repeating that the assumption that visual analysis is a more conservative procedure than statistical analysis is not supported by evidence.

Replication

We now consider the third strand we identified in the operant argument against the use of inferential statistics: The argument that it is replication rather than statistical analysis that best establishes the reliability of treatment effects. By *reliability,* we mean the extent to which the obtained difference between treatments was dependable (i.e., not produced by temporary fluctuations in random nuisance variables such as mood, memory, fortuitous environmental conditions, etc.). Traditionally, reliability has been evaluated by means of inferential statistics. Whether or not it is evaluated statistically, it is a necessary (although not sufficient) condition for internal validity. It allows us to infer that a systematic variable, as opposed to random nuisance variables, probably caused the observed difference between treatments. As we have seen, for the experiment to be internally valid, we must also be able to infer that the variable causing the difference between conditions was the one we manipulated (the independent variable) rather than some other systematic nuisance variable.

Within the operant tradition, there is a strong emphasis on *replication* as the means of establishing the reliability of a treatment effect. This goes beyond the recognition that, in both single-case and betweengroup experiments, replication is the ultimate arbiter of reliability. In a sense, single-case designs have replication built in, as in replications over

time in an ABAB design or across behaviors (or participants) in a multiple baseline design. It has been suggested that this can be viewed as an alternative to statistical analysis, as applied to traditional between-group designs. Kazdin (1976) made this point in a chapter that also contains a very useful summary of the arguments for and against the use of statistics in single-case experiments.

Although we are sympathetic with the emphasis on replication (and generalization) in single-case research, and agree that these criteria tend to be neglected in between-group research, built-in replications beg the question of whether each individual comparison in a single-case design can be regarded as reliable in the absence of a statistical test.

After all, visual analysts do use an explicit or implicit set of heuristics to make causal inferences, to which they attribute reliability, based on comparisons of dependent variable measures across treatment conditions (e.g., Parsonson & Baer, 1978). Here, the argument returns again to an emphasis on a search for large, easily discernable (clinical) effects in single-case research. If the effects reported were always obvious to visual inspection, there would be no problem. Unfortunately, the reality is not so straightforward. In their useful review of research looking at the reliability and validity of visual analysis of single-case data, Franklin, Gorman, et al. (1996) concluded that, even with experienced practitioners, sole reliance on visual analysis results in unacceptably high error rates. As we noted earlier, Parsonson and Baer (1978) argued that visual analysis is less sensitive than statistical analysis, thereby creating a bias in favor of only finding large and generalizable effects; that is, effects that are likely to be replicable and of clinical significance. As we also noted, however, Franklin, Gorman, et al. (1996) found no support for this prediction in their review.

In between-group research, there is certainly a need for a greater emphasis on direct replication to establish the reliability of effects identified by statistical analysis and for systematic replication (i.e., orderly variations in the circumstances of the original experiment) to establish generalizability of effects. This is particularly so in view of the usual reliance on the unsustainable assumption of random sampling from a population to which results can then be generalized (Edgington, 1996). In single-case research, there is no pretense of random sampling from a population and the need for replication to establish reliability and generalizability, whether within or between experiments, is immediately apparent. That does not, however, exempt the single-case researcher from the need to establish the reliability and internal validity of each individual effect identified.

CLINICAL AND STATISTICAL SIGNIFICANCE

In the preceding review of operant arguments against the statistical analysis of data in single-case experiments, we came to some general conclusions:

1. *Control.* In much human single-case research, the argument that control should replace statistical analysis is only sustainable when large (clinically significant) effects are the exclusive goal.
2. *Response-guided experimentation.* Random procedures (in place of or combined with response-guided procedures) should, in the interests of internal validity, be incorporated in single-case designs whenever possible. They are necessary to avoid a bias in favor of

finding spurious effects. Valid statistical tests of single-case designs are often unavailable in the absence of randomization procedures. The argument that this does not matter because statistical analysis is too sensitive (compared to visual analysis) to small (clinically insignificant) effects lacks empirical support.

3. *Replication.* The use of built-in replications in single-case designs does not exempt the researcher from the need to establish the reliability and internal validity of individual comparisons. The issue, once again, is whether evaluation should be on the basis of statistical or clinical significance.

In our consideration of control, response-guided experimentation, and replication, the operant argument against using inferential statistics seemed to turn on the issue of whether the appropriate criterion for the interpretation of individual comparisons is statistical or clinical significance. It is to this issue that we now turn.

There is no doubt that, in many single-case intervention studies, the goal is to make a clinical difference to the life of an individual or, more broadly, to ascertain whether a treatment is capable of making a clinical difference for at least one individual. Clinical significance is concerned with the importance of the change achieved and leads to a focus on powerful variables with large, robust effects. As Parsonson and Baer (1978) observed, these are likely to be variables with effects that are widely generalizable to suboptimal, nonlaboratory conditions.

Although there is no disputing that the principal goal of much single-case research is clinical significance, we need to recognize that there is an objective behavioral component to the clinical criterion. In addition to a requirement that people in the client's natural environment recognize that the original behavioral goal has been achieved (e.g., normal level of participation in tutorials), there is also a requirement for objective data showing behavioral change (e.g., increased frequency of contributions to group discussion in the postassertiveness training phase). Kazdin (1976) referred to the *experimental criterion,* which he identified as a concern with the reliability of the change produced by an intervention. We would add internal validity as part of the experimental criterion. This is because it is not enough to show that some systematic effect was reliable—we need to be clear that the effect was unambiguously attributable to the planned treatment and not some other uncontrolled systematic nuisance variable.

Traditionally, in between-group studies, the experimental criterion would be satisfied by control of potentially confounding variables and statistical evaluation of the difference between treatment groups. As we have seen, however, the use of statistics is not necessarily required to infer a reliable effect if variables are substantially under the experimenter's control or the effect is very large, and there is a strong tradition of using visual analysis in single-case studies. In the absence of a random allocation procedure, however, internal validity cannot be guaranteed, regardless of whether visual or statistical analysis is used. If assignment of treatments to available observation times is nonrandom, as in response-guided termination of a baseline phase, there will be a bias favoring low values for observations preceding the intervention point. This offers an alternative explanation of why values of observations in the treatment phase are higher than those in the baseline phase, if indeed they are. It seems to us that there is no satisfactory justification for failing to use a random allocation procedure in single-case studies when variables are not under tight control or the expected effect is not large.

There are certainly some circumstances when these conditions are unlikely to be met in single-case evaluations of treatment interventions. For example, investigations involving "new" variables are likely to be in this category. Whether or not effects are likely to be large may not be known in the early stages of a research program. Furthermore, effects that are initially small, but are persevered with because they are found to be reliable (i.e., statistically significant) may eventually turn out to contribute to clinical significance. This might occur when aspects (parameters) of interventions are modified in further research, or it might happen when a weak variable is used in conjunction with another variable. To rule out all small effects is in effect to deny a role to theory in the pursuit of large, clinically significant effects.

Sometimes, even though effects are potentially large, they may be difficult to evaluate by visual inspection because of the considerable intraparticipant variability present in the data. In this connection, it is worth reminding ourselves that the difficulty of evaluating causal hypotheses by visual inspection of graphed data is frequently underestimated, as Busk and Marascuilo's (1992) and Franklin, Gorman, et al.'s (1996) reviews of studies investigating the issue make clear. In particular, it will not always be possible to establish a stable baseline, yet there may still be real clinical gains even though the behavior in question fluctuates widely from one occasion to another. This is most likely when research is conducted in natural environments, where experimental control is limited. In these situations it may be necessary to use statistical analysis to determine whether the experimental criterion has been met and, in many cases, valid statistical analysis is dependent on the prior use of randomization procedures for the assignment of treatments to available observation times.

WHEN TO USE STATISTICAL ANALYSIS

We accept the preeminence of clinical significance in single-case (and small-*n*) human treatment intervention research. We think, however, that those who would exclude statistical analysis from all such research are limiting future possibilities of identifying new, clinically significant treatments. There is no need to choose between visual and statistical analysis (Edgington, 1984). There is no simple rule for deciding whether statistical analysis is appropriate for a particular study. It will always come down to the judgment of the researcher. We can only offer our conclusions about some of the considerations that the researcher might keep in mind when coming to that judgment.

1. The greater the experimental control, the less the need for statistics.
2. Much variability in the data suggests low experimental control.
3. Control tends to be lower in natural environments.
4. The larger the expected effect, the less the need for statistics.
5. Small effects and low control are often associated with new variables.
6. Replication is the final arbiter, but it is still necessary to establish the internal validity and reliability of each individual comparison.
7. Internal validity is compromised by response-guided procedures. Random allocation of treatments to times should be preferred unless variability is so low, and the effect so

clear, that there is little possibility of mistaking the systematic bias produced by nonrandom allocation procedures for a treatment effect.

8. When a random allocation procedure has been used, a valid randomization test will always be possible. Why not use one to check that visual analysis was not misleading with respect to the existence of a treatment effect? Evidence does not support the common belief that visual analysis is more conservative than statistical analysis.
9. People, even "experts," tend to be less good at making sound causal inferences from graphical data than they think they are.

Chapter 3
Approaches to Statistical Inference

We concluded in chapter 2 that there were likely to be some circumstances when clinical researchers using single-case or small-*n* designs would be well advised to consider the possibility of using statistical analysis as an adjunct to visual analysis of their data. We have argued that, for some experimental designs, the only valid and practical test is a randomization test. In others, alternative inferential statistics may provide approximations to the "ideal" randomization test solution. A statistical text by May, Masson, and Hunter (1990) treats randomization tests and classical parametric tests in parallel and considers the relation between them. This is exceptional, however. Randomization tests receive no mention in most standard statistical texts, and one might ask why, if they are so appealing, do they receive so little coverage. To answer the question, we need to take a look at the historical developments that have led to this state of affairs.

RANDOMIZATION TESTS

The fundamental problem of statistical inference is the same as that facing a jury in a court of law: How should the evidence be weighed? As was pointed out in a classic work by Jeffreys (1939), traditional (deductive) logic has nothing to contribute to this problem. Several approaches to it have been developed since the 1920s, beginning with the work of Fisher (1935). One of his methods was the randomization test. The great advantage of this was that no assumptions were made about the data: Provided that the experiment was carried out correctly, with treatments randomly assigned to experimental units (e.g., to participants or available observation times), nothing more was needed. It was not even necessary to assume that experimental units were random samples from some population, although it was Pitman (1937) rather than Fisher who demonstrated this characteristic of randomization tests.

Unfortunately, randomization tests were not easy to classify and often had to be thought out afresh for a new problem. Even worse, the calculations were lengthy, tedious, and very difficult to automate. Fisher was a virtuoso performer on the primitive mechanical calculators of the time, but even he could make little use of randomization tests. In fact, they were little used until the widespread availability of fast computing made it feasible to make numerous reorderings and resamplings of data. In the 1970s, for example, Efron (1982) used the computing power then available to extend the randomization test approach to other resampling methods, such as the bootstrap. He commented that if we made as good use of our computing equipment as Fisher made of his, statistics would be in a very different place by now.

THE CLASSICAL THEORY

The early statisticians had to compensate for lack of computing power by making the best possible use of analytical tools, and statistical theory developed a strongly mathematical approach. Progress was made by using families of distributions with "nice" mathematical properties that between them could be used to approximate many real problems. The normal distribution and its derivatives chi-square, *t*, and *F* were exploited, notably by Fisher and "Student" (Gossett). In the 1930s, Neyman and Pearson (1967) brought their formidable powers to bear on the problem and they extended and systematized this work in a series of joint papers. For the first time there was a coherent theory of statistical hypothesis testing, with clear assumptions and methods, definitions of power, sufficiency and efficiency, and an understanding of the reasons for using the tests that remain the most familiar today. The orderliness of this approach and the lack of understandable alternatives assured it wide acceptance. However it was not without problems and it provoked controversy among statisticians from the beginning.

One difficulty is that these methods provide excellent answers to questions that are not those most researchers are asking. What you really want is to be able to make a probability statement about the hypothesis given your data. What you get is a probability statement about your data given the null hypothesis, which is usually not even the one you are interested in. Experiments have shown that people find conditional probability very hard to understand, so what can they be expected to make of using prob(data given H_0) when what they want is prob(H given data)? The reason we ended up with classical (Neyman-Pearson) statistics is that no one could solve the problems as posed, so they used an oblique approach that does throw some light on the situation, but does not focus on the question of most interest. Generations of students have struggled with this confusing approach to hypothesis testing and our impression is that it is the best students who have the most trouble. They are confused by getting a good answer to a question not asked, whereas weaker students, being less clear what the question was, do not worry.

The focus on the null hypothesis arose because it is a simple hypothesis, in the sense that it refers to a single value (zero difference), and the families of distributions (such as the normal distribution) to which it was allied were tractable and provided plausible models for many real problems. The research hypothesis, on the other hand, is complex in that it refers to a range of values (nonzero differences). Even when no prior assumptions are made about distributions, as with nonparametric statistics (including randomization tests) the research hypothesis remains complex because the size (i.e., range of values) of the hypothesized treatment effect is generally unspecified. Hence, the focus remains on the null hypothesis in nonparametric tests.

In addition to the fundamental philosophical difficulty relating to the null hypothesis, the necessity of relying on mathematical analysis means that restrictive assumptions about the sampling process and the distributions are needed. In addition to the requirement, common to all hypothesis-testing methods, that observations (strictly, the "error" components of observations) must be independent of one another, the Neyman-Pearson theory requires (a) random sampling from properly defined populations, (b) usually normal distribution of the dependent variable, and (c) equality of variances in all populations. We have already noted

that the random sampling requirement is hardly ever met. In addition, although the equality of variance requirement is generally recognized with respect to the *t* test, it seems often to be conveniently forgotten in ANOVAs, to which it also applies. Although it is frequently claimed that parametric tests are robust (i.e., statistical decisions are not invalidated) in the face of moderate departures from these assumptions, more extreme violations of assumptions are not infrequent in practice (Wilcox, 1987). Various modifications to the statistical procedures have been developed to deal with particular violations of assumptions, although these are not highly accessible to the average consumer of statistical tests.

BAYESIAN INFERENCE

The Neyman-Pearson approach was criticized by statisticians and philosophers from the beginning, and attempts were made to find a more satisfying approach to the whole problem of inference. A helpful line of enquiry, pioneered by Jeffreys (1939) and developed by many authors, was based on Bayes' Theorem, a theorem in probability known since the 17th century. By the mid-1960s this work was sufficiently well developed for Lindley (1965) to write an accessible textbook on Bayesian inference. Although this approach does allow you to look at the probability you want (i.e., the probability of the research hypothesis being correct), it makes even more lavish use of not-always-plausible assumptions. In particular, it has a whole clutch of mystifying assumptions about the nature of belief, and has to get people to act on these assumptions and produce prior probabilities. These are probabilities that represent the beliefs of the experimenter before any data have been collected. Not only is there the same need for tractable distributions encountered by Neyman-Pearson; there is also a requirement for the experimenter to specify a (tractable) prior probability distribution over the possible outcome values. Experiments show that this is hard for people to do, and within the Bayesian school opinions are sharply divided about how it should be done. Lindley favored vague priors, which in practice make all or many values equally probable or nearly equally probable. Jeffreys, on the other hand, maintained that no one would perform experiments without having strong prior belief in a particular value. What it seems to come down to is a willingness to acknowledge a place for subjectivity in statistical analysis. It all seems rather a long way from the data and, among researchers, there has never been a large following ready to abandon objectivity.

RANDOMIZATION TESTS REVISITED

Unless we are willing to go down the Bayesian route of subjectivity, and some researchers clearly are, we have to live with testing the null hypothesis instead of the research hypothesis. We have to acknowledge that we simply do not know how to infer the general (the research hypothesis) from the particular (the data) without abandoning objectivity. It is no longer necessary, however, to accept the additional limitations of the classical approach. Specifically, with our current computational power, there is no necessity to be constrained by a requirement for tractable distributions that are amenable to analytic solution based on a random sampling assumption.

Randomization tests take it that all the information we have about the distribution from which our data come is in the data. In this approach, we accept that the data are our best, indeed our only, estimate of the distribution. Then we work out the distribution of the test statistic under the null hypothesis, using computational power. This differs from the classical approach, which uses mathematical analysis to work out the distribution of the test statistic from the assumed distribution from which the data are assumed (usually with no justification) to be a random sample. It is also the case that the actual statistic used in Randomization tests may be much simpler mathematically and "closer" to the hypothesis than that used in classical tests (e.g., the "difference between means" in place of *t).* The final step in both approaches is the same: We look to see where the actual test statistic is in its distribution and, if it is in a tail, we conclude that random variation is not a good explanation. So the two assumptions we evade are the nature of the distribution and random sampling from it. By removing some of the unreality associated with these two assumptions, we put a lot more weight on the data.

Classical (parametric) tests (e.g., *t* test, ANOVA, etc.) and the familiar (nonparametric) rank test alternatives (e.g., Mann-Whitney, Wilcoxon, etc.) may be regarded as giving asymptotic approximations to the exact probabilities provided by randomization tests. As we observed in chapter 1, for most multiple-participant designs the approximations are reasonably satisfactory, and the prospect of jettisoning the restrictive assumptions appears to provide insufficient incentive to switch to randomization tests now that the computational power exists to make them feasible. For single-case and very small-*n* designs, however, they should often be the procedure of choice when a statistical analysis is desired.

ASSUMPTIONS OF RANDOMIZATION TESTS

As we noted in chapter 1, the principal assumption that is required for a randomization test is that the data arrangements (reorderings of the data) used in the test match the procedure used for random assignment of treatments to observations. That is, each reordering of the data must be one of the orderings that could have occurred as a result of the random assignment procedure. There are circumstances in which it may be reasonable to relax this assumption, but we leave discussion of that issue to a later chapter. For the time being we accept that this assumption must be met. We have asserted that randomization tests do not require any assumption about the form of the distribution, but that assertion is in need of clarification. Also, possible violations of the assumption of independence of observations must be considered.

Distributional Assumptions

We have said that randomization tests make no prior assumptions about the form of the distribution from which our data come, but it is also true that the shapes of distributions may affect the statistical decisions made. Nonparametric tests, including randomization tests, will be sensitive to any difference between distributions, not just the difference between means, which is the usual focus of interest. Indeed, there are some nonparametric tests, such as the Kolmogorov-Smirnov test (Siegel & Castellan, 1988) that are explicitly

designed to test for any difference between distributions. The central point here is that, if you are testing a simple hypothesis, you have to assume there are no other differences between the groups you are comparing, in statistics as elsewhere. If two distributions have different shapes, observing that their means are different is hard to interpret because it is an inadequate account of the difference. This is a familiar problem in parametric statistics, where if you find an effect at one level, effects at simpler levels cannot be interpreted. Thus, it is widely understood that a significant interaction in an ANOVA creates difficulties for the interpretation of main effects in the analysis. Randomization tests make no assumptions about the forms of distributions (and they are not assumed to be the same for all experimental groups) so they will pick up any differences between distributions as well as differences we are looking for (e.g., between means). A difference in shape is analogous to the more complex (interaction) effect in the ANOVA, and therefore implies that interpretation of the simpler (mean difference) effect will be problematic.

An example may help to clarify the argument. Suppose an intervention (e.g., a relaxation technique) in a phase design leaves most observations (e.g., amount of participation in group discussions) in the postintervention phase unchanged but results in a few much higher scores. This will increase the mean for the treatment group, but the treatment distribution is also skewed, so it has a different shape from the control group. You would want to know this, because it might suggest that the intervention was interacting with an uncontrolled variable (e.g., discussion topic) and you might wish to control candidate variables in future studies. A randomization test using the difference between means as the test statistic would pick up the effect. Saying it changed the mean would obscure the true nature of the effect, of course, but visual inspection of the data may suggest a hypothesis about what the true effect may have been, which further research may test directly. This is a good example of the potential for a constructive combination of statistical and visual analysis. Gorman and Allison (1996) asserted that, "although randomization tests for differences in central tendency can be used with distributions of any shape, the two (or more) treatment conditions being compared must have the *same* shape and variance" (p. 171). Although it is true that if this condition is not met it will be impossible to identify unequivocally the nature of the effect, when the test is used in conjunction with detailed inspection of the data it may still yield useful information, as in the preceding example.

It is also worth considering how important Gorman and Allison's (1996) requirement is in practice. It may help to examine a very extreme scenario. Suppose one treatment gave an exponential score distribution (i.e., a very skewed distribution) and the other treatment (or control) gave a symmetrical distribution, but both had the same mean. Then, if you had just four observations from each, it can be shown that the chance that all four would be on the left of the mean would be nearly 0.15 for the exponential distribution and 0.0625 for the symmetrical distribution. The chance of a complete separation, with all of the scores in the exponential distribution to the left of the mean and all of the scores in the symmetrical distribution to the right of the mean, would be $0.15 \times 0.0625 = 0.0093$. The randomization test would give a probability of $4!4!/8! = 0.014$ for complete separation (the formula is explained in chap. 4). This seems very close, particularly because the probability based on the known distributions is "too low" because it specifies opposite sides of the mean as well as complete separation (i.e., all of one followed by all of the other). Although we do not disagree with Gorman and Allison's (1996) equality requirement for a safe inference about

a difference in means based on a randomization test, we do wonder about its importance in practice. Furthermore, other approaches make assumptions that are as restrictive (rank tests) or more restrictive (parametric tests).

Autocorrelation

It is known that serial dependency often exists in time-series data. That is, when repeated observations are made on the same participant, there is a likelihood that errors of measurement (residuals) associated with scores at one data point may be predictive of errors at other points in the series that follow. Generally, the closer one observation is to another in the series, the more highly correlated the residuals of scores are likely to be. This is referred to as *autocorrelation*. The correlation between residuals of scores that are adjacent in a time-series (i.e., Observationl paired with Observation 2, Observation 2 paired with Observation 3, etc.) are described as a lag-1 autocorrelation. The correlation between residuals of scores that have one intervening observation (i.e., Observation 1 with Observation 3, Observation 2 with Observation 4, etc.) will be a lag-2 autocorrelation, and so on. Generally, except where there are cyclical variations, autocorrelations tend to be smaller at higher lags.

The presence of autocorrelation violates the assumption of the independence of observations required for all hypothesis testing methods. It is known that the presence of positive autocorrelation in parametric tests results in underestimation of probabilities. That is, there will be a endency to find more significant effects when the null hypothesis is really true (Type I errors) than is indicated by the reported alpha level. To put it differently, effects that are reported as significant at, say, the 0.01 level will really only be significant at a higher level, such as the 0.05 level. It has been suggested that randomization tests applied to serially dependent data in single-case designs are immune from the effects of serial dependency because the test "is based on the null hypothesis that there would be identical responses across occasions if the conditions were presented in a different order" (Barlow & Hersen, 1984, p. 306). Although this argument might be sustainable in relation to alternating designs (where times are randomly assigned to treatments), it is not persuasive when applied to phase designs (where treatments are blocked within phases and only the point of intervention is randomly determined) unless, as Kazdin (1984) and Levin et al. (1978) recognized, phase means rather than individual observations are treated as the experimental units to be analyzed. Edgington (1984) claimed that "serial correlation in a temporal series of observations from a single subject violates the assumption of independence underlying the F table and other parametric significance tables" (p. 92). But, as Good (1994) observed, the independence of observations is a requirement for all hypothesis testing methods. In his discussion of weather research in support of randomization tests, Edgington (1984, 1995) seemed to be mixing the autocorrelation issue with the random sampling issue. It is undoubtedly the case that random sampling of temporal sequences would provide a poor model of weather systems, but it remains questionable whether a randomization test will deal with the autocorrelation inherent in weather predictions (e.g., in cloud seeding experiments) unless observation times are randomly assigned to the appropriate experimental units. The point is that the randomization test must map onto the randomization procedure: An incomplete randomization procedure (as in random determination of an intervention point in a phase design) will not preserve the researcher from the effects of autocorrelation within the appropriate experimental units (phases, in this case).

The presence of autocorrelation in phase designs takes us back to the issue of when to end a baseline phase. If positive autocorrelation exists in the baseline phase, a run of similar measurements (i.e., meeting a stability criterion) may be more likely than a random probability model would suggest. In that case, the apparent stability might reflect the non-independence of the observations rather than nonvariability of the behavior. An analogy may help to clarify the argument. The best prediction for tomorrow's weather, based only on previous observations of the weather, is that it will be like today's. Assume that the probability of tomorrow's weather being the same as today's is 0.6. Suppose there has been a dry spell and we are anxious for it to rain. The probability of rain within the next 5 days is ($1–0.6^5=0.92$), so if the local rainmaker does a rain dance it will most likely rain soon and he will take the credit. Of course, rain would be equally likely in the few days following 1 dry day, but any competent rainmaker will know that it will be much more impressive if he does his rain dance after a run of dry days. When the timing of an intervention is guided by baseline responses, it may share some of the characteristics of a rain dance, and the similarity will be most pronounced in the presence of autocorrelation.

Returning to the question of whether autocorrelation results in too many significant effects in single-case designs, there are some data to help us. Gorman and Allison (1996) reported preliminary results of an empirical investigation of the effects of autocorrelation on the probability values obtained. They reported that, in simulations of an AB phase design with four to six observations per phase, the greater the lag-1 residual autocorrelations (the range was from 0.0–0.90), the greater the Type I error rate. As there was no randomization within phases and individual observations comprised the unit of analysis, this does not seem particularly surprising. They expressed surprise that the effect of autocorrelations was greater as the number of observations was increased. Again, we find this unsurprising. An example may help to explain why. If you imagine making observations every hour instead of every day, so you have maybe 10 times as many, it is easy to see the results would look more impressive if you did not realize there was not really any more information. The higher the autocorrelation, the less data you have "really" Bigger samples may therefore be expected to amplify the effect.

Although the effects of autocorrelation were smaller in a simulation of an alternating treatment design, they were still in evidence and it is our turn to be surprised. We are puzzled as to what the explanation of the effect could be when there is random assignment of observation times to treatments. That is, the randomization procedure is applied to the unit of analysis, which is the treatment score at each observation time. This preliminary finding is clearly in need of replication. We can conclude, however, that autocorrelation may be a problem for at least some single-case designs.

How is the problem to be tackled? Gorman and Allison (1996) reported that they were attempting to tabulate actual and nominal Type I error rates for a range of sample sizes and degrees of autocorrelation so researchers can apply appropriate corrections to significance values. We doubt that this will provide a satisfactory solution in view of the imprecision of estimates of autocorrelation based on a small number of observations. Another suggestion, that alternating treatment designs be used when possible, seems sensible, but there may be rather few occasions when a switch from a phase approach to an alternating treatment approach will prove practical. They also suggested adoption of a conservative significance level, which seems a reasonable precaution when positive autocorrelation is suspected.

The best advice, probably, is to take steps to reduce the likelihood of autocorrelation occurring in the first place. An obvious step, as Levin et al. (1978) noted, is to maximize the interobservation interval. Another opportunity for reducing autocorrelation concerns the way in which measurements are obtained. When the same individual makes repeated measurements (e.g., ratings of social effectiveness) serial dependency within the measures is quite likely. Such serial dependency would be far less likely if different individuals were randomly assigned to observation periods, with each making a single measurement. Suitable rater training, together with checks on interrater reliability could be employed to prevent unnecessary variability being introduced into the data. With regard to the variability issue, it needs to be recognized that, if autocorrelation exists due to a single rater making all of the measurements, the reduced variability will be illusory. There will be fewer "real" (i.e., independent) measurements than the nominal number reported. Finally, the more objective the measurements can be made, the less autocorrelation there is likely to be and, of course, the less need for the measurements to be made by different individuals.

THE CURRENT SUITABILITY OF RANDOMIZATION TESTS

Returning to the question asked at the beginning of the chapter, we can now summarize what we see as the reasons for the relative neglect of the randomization test approach to statistical inference and why we believe the time has come to end this neglect.

1. Analytical power, as exemplified in classical tests (and likelihood tests), was necessary when computational power was lacking, but this is no longer the situation. Randomization tests substitute computational power for analytical power. This has three advantages:
    - They can use a statistic (e.g., the difference between means) that is both simpler and more transparently related to the hypothesis.
    - They use an actual distribution of the statistic rather than a hypothetical distribution that is known only through mathematical analysis.
    - Parametric assumptions about the form of score distributions are not required by randomization tests. Although the actual forms of distributions affect the interpretation of test results, this is true of all hypothesis testing approaches.
2. Classical tests (and likelihood tests) assume random sampling from well-defined populations of interest. In principle, this means that significant differences between samples can be generalized to the population. In practice, groups are almost always formed by random assignment of available participants. This is consistent with the randomization requirements for randomization tests, including those for single-case designs, where the random assignment is of available times to treatments. As far as generalization is concerned, in the absence of random sampling, classical tests are in precisely the same position as randomization tests, which have the merit of not making unsustainable sampling assumptions. Without the baggage of the untenable random sampling assumption, randomization tests focus on the kind of conclusion sought by the researcher: Is the difference between scores in different conditions attributable to chance or to something systematic?

3. When the random sampling assumption is not met, classical tests (and likelihood tests and nonparametric rank tests) provide approximations to the exact probabilities provided by randomization tests. These approximations serve reasonably well for large-*n* designs but, for very small-*n* and single-case designs, exact probabilities are preferable. Indeed, there are no classical or standard rank tests available for many possible single-case designs.
4. The problem of autocorrelation arises in all time-series designs. It is probably less of a problem for randomization tests than for classical tests, particularly for designs in which there is full random assignment of treatments to observation times, as in alternating treatment designs. ARIMA models can remove autocorrelation, but they are complex and require many more observations than are usual in single-case studies. The best advice is to avoid or reduce autocorrelation by suitable spacing of measurements and use of multiple raters.
5. Although the Bayesian approach deals directly with the hypothesis of interest, rather than the null hypothesis, it has the drawback of requiring estimates of prior probabilities based on the subjective beliefs of the researcher, in addition to the usual parametric assumptions. Those who are uneasy about subjectivity in statistical analysis will likely prefer the objectivity of the randomization test approach.

Chapter 4
Principles of Randomization Tests

In the preceding three chapters we have tried to set the scene for a detailed account of how randomization tests work. In chapter 1, we considered when it would be appropriate to use a randomization test, assuming that a statistical test of some kind was wanted. In chapter 2, we considered the case for using statistical tests at all in clinical research, and the related issue of using randomization procedures instead of response-guided procedures in clinical research. In chapter 3, we considered why randomization tests have been so little used and why we believe that their time has come, especially in relation to single-case experiments. In this chapter we use several examples to illustrate how randomization tests work. We begin with a hypothetical experiment described by Fisher (1935).

THE LADY TASTING TEA EXPERIMENT

Fisher (1935) described a hypothetical experiment designed to test a lady's claim that she could discriminate by taste between tea made with milk (M) poured first and tea made with tea (T) poured first. In this imaginary experiment, the lady was presented with four cups of tea with milk poured first and four cups with tea poured first, with their order of presentation randomized. The number of ways in which the eight cups can be ordered with respect to tea or milk poured first is an example of *combinations*. This is the number of possible orderings when we do not differentiate between cups with tea made in the same way. If we wanted to know the total number of possible orderings of all eight cups, that would be an example of *permutations*. It is quite confusing, as Gorman and Allison (1996) noted, that the reorderings carried out in randomization tests are frequently referred to as permutations even when, as in this case, they are actually combinations. We have followed Gorman and Allison in using the more neutral term *arrangements* (or *rearrangements*) to cover all types of reorderings.

The number of possible arrangements of eight things taken four at a time is calculated using factorials of numbers, which are written as the number followed by an exclamation mark. Thus, four factorial is written as 4!, which means 4 multiplied by all of the positive digits below it (i.e.,4×3×2×1). The number of possible combinations of eight cups of tea, with four of Type M and four of Type T, is given by 8!/(4!4!)=(8×7× 6×5)/(4×3×2×1)=70. This is a well-known result, but explanation follows for readers who are unfamiliar with it but wish to understand its derivation.

There are 4! (i.e., 4×3×2×1) ways to arrange four different objects in a row. There are 4 ways to choose the first object and 3 ways to choose the second, and they can be in any combination, so we have 4× 3 ways to choose the first two. There are 2 ways to choose the third, so 4 ×3×2 ways to choose the first three. Finally, there is only one left to occupy fourth place. Similarly, there are 8! (i.e.., 8×7×6×5×4×3×2×1) ways to arrange 8 different objects in a row, but if 4 of the 8 are the same, then any of the 4! arrangements of these among themselves will give arrangements of the 8 that are the same. So there are 8!/4! different arrangements. If the remaining 4 are also alike, but different from the first 4, we have 8!/(4!4!) different arrangements of 8 items, 4 each of 2 kinds.

Returning to the hypothetical experiment, one of the 70 possible orders for presentation of the cups was selected by means of randomizing the order. The null hypothesis in this experiment is that the lady is unable to tell the difference between cups with tea poured first and cups with milk poured first. If the null hypothesis is true, that means that her sequence of decisions (T, M, M, T, M, etc.) will be totally unrelated to what is actually in the cups, so if the cups had been presented in some other order, her sequence of Ms and Ts would have been the same. So, we ask how many are correct for the actual order in which she received the cups, and how many would be correct for each of the other possible orders in which they could have been presented. Assume that she had the top possible score, with all eight correctly identified. Either her correct identification of all eight cups happened by chance (with a probability of 1/70) or she really can tell the difference and the null hypothesis is false. We conclude, therefore, that the result is significant at the $100\times1/70=1.43\%$ level (i.e., $p=0.0143$ and $p<0.05$).

EXAMPLES WITH SCORES AS DATA

Fisher's illustration of the principle underlying randomization tests is an example of a single-case *alternating treatment* design. In this example, the test statistic is the number of correct cups. We now provide another example of this design in which the data are scores rather than nominal values (i.e., right or wrong). These designs are analogous to a standard, completely randomized design, in which two alternative treatments are randomly assigned to participants. There are two important differences: First, the assumption of random sampling from populations can be unequivocally abandoned and, second, instead of assigning treatments to different participants, treatments are randomly assigned to the observation times available for a single participant. Both of these differences are important. We use an example, adapted from May et al. (1990), to illustrate how a statistic is computed when the data are scores and how a region for rejection of the null hypothesis is established. We begin with an example that is small enough to allow us to display all of the (20) possible arrangements of the data and then we will move on to an example that is the same as Fisher's (i.e., 70 arrangements) except that the data are scores.

Alternating Treatments (3 Observation Times for Each of 2 Treatments)

Suppose that we are interested in the effects of praise on the number of minutes a child stays on task when the praise is given by an age peer (P) or by an adult (A). Suppose, also, that we expect longer on-task behavior when praise is given by a peer. We plan to administer a task on six occasions, so three P and three A treatments are randomly assigned to the six task occasions. For the sake of simplicity, we will assume that the scores obtained are 1, 2, 3, 4, 5, and 6 min. We wait a bit before considering to which treatment condition each score belongs. In the meantime, there are $6!/(3!3!)=20$ ways that the six scores could be randomly assigned to the two treatments and these possible assignments are listed in Table 4.1.

Also in Table 4.1, we have given the difference between P and A means for each of the 20 arrangements. Now we consider how the scores were actually distributed between the two treatment conditions. Assume that the actual scores for the P condition were 3, 5, and 6 (M= 4.67) and those for the A condition were 1, 2, and 4 (M=2.33). The difference between these means is 2.34 and it is in the predicted direction. The probability of an arrangement of scores occurring by chance (i.e., when the null hypothesis is true) to give a positive difference of this size or bigger is 2 out of the 20 possible arrangements (i.e., the first 2 arrange ments in the table). As this probability is greater than 0.05 (i.e., p=0.1), the difference

TABLE 4.1 All Possible Arrangements of Six On-Task Scores Assigned to the Two Treatment Conditions (P and A)

Arrangements	*A*	*P*	*Mean P–Mean A*	=	*Difference*
1	1, 2, 3	4, 5, 6	5.00−2.00	=	+3.00
2	1, 2, 4	3, 5, 6	4.67–2.33	=	+2.34
3	1, 3, 4	2, 5, 6	4.33–2.67	=	+1.66
4	1, 2, 5	3, 4, 6	4.33–2.67	=	+1.66
5	1, 2, 6	3, 4, 5	4.00–3.00	=	+1.00
6	1, 3, 5	2, 4, 6	4.00–3.00	=	+1.00
7	2, 3, 4	1, 5, 6	4.00–3.00	=	+1.00
8	1, 3, 6	2, 4, 5	3.33–3.33	=	+0.33
9	1, 4, 5	2, 3, 6	3.67–3.33	=	+0.33
10	2, 3, 5	1, 4, 6	3.67–3.33	=	+0.33
11	2, 3, 6	1, 4, 5	3.33–3.67	=	−0.33
12	2, 4, 5	1, 3, 6	3.33–3.67	=	−0.33
13	1, 4, 6	2, 3, 5	3.33–3.67	=	−0.33
14	1, 5, 6	2, 3, 4	3.00–4.00	=	−1.00
15	2, 4, 6	1, 3, 5	3.00–4.00	=	−1.00
16	3, 4, 5	1, 2, 6	3.00–4.00	=	−1.00
17	2, 5, 6	1, 3, 4	2.67–4.33	=	−1.66
18	3, 4, 6	1, 2, 5	2.67–4.33	=	−1.66
19	3, 5, 6	1, 2, 4	2.33–4.67	=	−2, 34
20	4, 5, 6	1, 2, 3	2.00–5.00	=	−3.00

is not statistically significant (i.e., p>0.05) in a randomization test of the directional (one-tailed) prediction. Now suppose that the actual scores were 4, 5, and 6 (M=5.00) for the P condition and 1, 2, and 3 (M=2.00) for the A condition. The difference between the means would be 3.00 in the predicted direction and the probability of a positive difference as big as this arising from a random arrangement of the scores would be 1 out of 20 (i.e., the

first arrangement in the table). This difference would therefore be statistically significant ($p<0.05$) in a one-tailed randomization test. It may be noted that only the most exrreme difference between means could reach significance in this example, where there are only 20 possible arrangements of the data. It is also the case that a significant difference is impossible in a two-tailed test, because, for any arrangement that gives the largest possible difference, there will be a reverse arrangement that will give the same difference in a negative direction (i.e., the 1st and 20th arrangements in the table).

For experiments where there are many more possible arrangements, there will be a range of differences between conditions that occur with a probability that places them in the *region of rejection* of the null hypothesis. With a large number of possible arrangements, however, it will generally be impractical to inspect the full complement of arrangements to identify the region of rejection, as we did in Table 4.1. It is helpful then to identify *classes of outcome,* which are sets of arrangements that result in the same mean difference. We illustrate this with the data in Table 4.1. In this example, the 20 outcomes can be collapsed into 10 outcome classes, shown in Table 4.2. Also shown is the frequency of outcome within each class and the probability of an outcome in that class occurring. This probability is simply the sum of the probabilities of the individual outcomes within the class. There are, for example, three arrangements of the data that yield a mean difference of −1.00, so the probability of an outcome in this class is 0.05+0.05+0.05=0.15. Finally, the cumulative probabilities for classes of outcomes in the predicted direction are given. It is clear that only the cumulative probability for a positive difference of +3.00 is in the 0.05 region of rejection for a directional (one-tailed) hypothesis.

TABLE 4.2 Outcome Classes of Possible Arrangements of the Data in Table 4.1

Outcome Difference Between Means	***Frequency***	***Probability***	***Cumulative Probability***	
+3.00	1	0.05	0.05	<Region of rejection with a=.05 and a directional hypothesis
+2.34	1	0.05	0.10	
+1.66	2	0.10	0.20	
+1.00	3	0.15	0.35	
+0.33	3	0.15	0.50	
−0.33	3	0.15	0.65	
−1.00	3	0.15	0.80	
−1.66	2	0.10	0.90	
−2.34	1	0.05	0.95	
−3.00	1	0.05	1.00	
Sum 0.00	20	1.00		

Alternating Treatments (4 Observation Times for Each of 2 Treatments)

We now provide an illustration of a randomization test on simulated data from the same design but with a larger number of possible arrangements. In this example, there are four observation periods for each of the treatment conditions, as in Fisher's lady tasting tea example. That gives 8!/(4!4!)=70 possible arrangements. Four P and four A treatments are randomly assigned to the eight task occasions. As in the previous example, scores are generated instead of binary decisions. Rather than listing all of the arrangements and the differences between means that they produce, in Table 4.3 we list just the outcome classes, their frequencies, probabilities, and cumulative probabilities, as we did in Table 4.2. In this example, we are assuming that the actual scores obtained were 5, 6, 6, and 6 (M=5.75) for the P condition and 5, 4, 4, and 4 (M=4.25) for the A condition. All possible classes of arrangement outcomes for the eight scores are shown in Table 4.3, together with their frequencies, probabilities, and cumulative probabilities. Again, the prediction is that scores will be higher in the P condition.

It can be seen from Table 4.3 that there are two ways of getting the most extreme positive difference of +1.5. If we identify the scores (5, 6, 6, 6, 5, 4, 4, 4) as first to eighth (i.e., 5=1st, 6=2nd, 6=3rd, 6=4th, 5=5th, 4=6th, 4 =7th, 4=8th), the two ways of getting a difference of +1.5 are (a) with the 1st, 2nd, 3rd, and 4th scores in the P condition, or (b) with the 2nd, 3rd, 4th, and 5th scores in the P condition (i.e., the three scores of 6, plus one of the two scores of 5). As the cumulative probability of this outcome class is 0.029, if either of these arrangements was the actual way scores were distributed between the P and A conditions, the obtained difference would be in the region of rejection (i.e., p<0.05) for a directional hypothesis, P> A. As we said that the scores in the P condition were 5, 6, 6, and 6 and those in the A condition were 5, 4, 4, and 4, the randomization test finds the difference to be statistically significant. In other words, the probability of a randomly selected arrangement of the scores into a P pile and an A pile producing a positive difference (P−A) at least as great as that actually obtained is less than 0.05, so we reject the null hypothesis as being "unlikely" to be true. It is also clear from Table 4.3 that if we had a nondirectional hypothesis (P≠A), the combined probability of the most extreme positive (+1.5) and the most extreme negative (−1.5) outcome classes is 0.0286+0.0286=0.0572, so the obtained absolute difference of 1.5 would not be significant in a two-tailed randomization test.

A Phase (Baseline—Treatment) Design

We complete our illustration of the principles underlying randomization tests with an example of a phase design in which a randomization procedure has been introduced. In this example, the randomization procedure is more limited, but is sufficient to permit the application of a randomization test to the data. We use an example provided by Edgington (1995). He described a hypothetical experiment in which the effectiveness of a reinforcement procedure for increasing a target behavior is investigated. Twenty treatment blocks are specified, with frequency of the target behavior being recorded in each. Once the treatment

TABLE 4.3 Possible Outcome Classes of Arrangements for Two Groups of Four Scores

Outcome Differences Between Means	*Frequency*	*Probability*	*Cumulative Probability*	
+1.5	2	0.0286	0.0286	<Region of rejection with a=.05 and a directional hypothesis
+1.0	6	0.0857	0.1143	
+0.5	18	0.2571	0.3714	
0.0	18	0.2571	0.6285	
−0.5	18	0.2571	0.8856	
−1.0	6	0.0857	0.9713	
−1.5	2	0.0286	0.9999a	
Sum	0.00	70	0.9999a	

"This value departs from 1.000 due to rounding error.

is introduced, it remains in effect in all subsequent blocks. To ensure that there will not be too few observations in either the baseline (B) or treatment (T) phase, the experimenter stipulates that each phase must contain at least five observations. Within this constraint, the experimenter randomly selects a starting point for the introduction of the intervention. That is, one of the blocks from 6 to 16 inclusive is randomly selected as the first intervention block. There are therefore 11 possible assignments of the treatment intervention to the blocks. Suppose the eighth block was selected for the commencement of the treatment and the results were as follows, with the treatment blocks shown in bold:

Block	1	2	3	4	5	6	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Data	2	3	4	3	2	3	4	**8**	**9**	**8**	**9**	**10**	**8**	**9**	**9**	**8**	**10**	**9**	**8**	**8**

The difference between baseline and treatment means (T–B) is a suitable statistic for a one-tailed test where the prediction is T>B. There are 11 possible arrangements of the data, corresponding to the 11 possible starting points for the intervention. The statistic (T–B) is computed for each of these arrangements. The value of the test statistic for the first arrangement is the mean of the first 5 blocks subtracted from the mean of the remaining 15 (120/15−14/5=5.20). The value for the second arrangement is 117/14−17/6=5.52. The value of the third arrangement, which is the one that was randomly selected to divide baseline from treatment blocks, is 113/13−21/7=5.69. Computation of the values of the statistic for the remaining arrangements would show that only the actual data division results in a statistic value as large as 5.69. Therefore the probability of a random arrangement of data divisions producing a statistic at least as large as the obtained value is 1/11=0.091. Clearly, this is the lowest probability that could be obtained with 20 blocks and at least 5 in each phase. For one-tailed statistical significance at the 5% level to be a possibility, there would need to be a minimum of 20 possible intervention points, from which one is selected at random.

Although the absolute difference between B and T phases may be a suitable statistic for a two-tailed test, it lacks sensitivity when there is a consistent upward or downward trend over the duration of the study. Edgington (1995) suggested that, in such circumstances, the analysis of covariance statistic F can be used in place of the absolute difference between phase means, with block numbers (1, 2, 3, 4, etc., up to 19, 20) entered as covariate.

GENERALIZED APPLICATION OF PRINCIPLES OF RANDOMIZATION TESTS

The preceding examples should make it clear that design and hypothesis testing considerations are inextricably linked in the randomization test approach. The general requirements for a randomization test apply equally to single-case designs and to small-*n* and large-*n* group designs. The requirements are as follows.

A Randomization Procedure

It is necessary to randomize some aspect of the experimental design. For an AB single-case design, we have seen that this might be, for example, random selection of the point at which to switch from baseline to intervention phase or, in an alternating treatment design, random assignment of treatments to available times. It is worth emphasizing that some form of random assignment procedure is a necessary prerequisite for doing a fully valid randomization test, and the way the random assignment has been done determines the precise form of the randomization test that follows.

Selection of a Test Statistic

A suitable test statistic is calculated from the data obtained. In the lady tasting tea example, it was the number of correct cups. In the other examples, the statistic was the difference between means in treatment and control conditions, although we saw that F from an ANCOVA was an alternative where data trends might make an absolute difference between means insensitive to treatment effects. It is often possible to use a statistic such as t or F that is commonly used to test the difference between means in parametric tests. Alternatively, some other convenient statistic such as the sum of squared deviations (SS) that is normally computed along the way to obtaining a standard statistic such as F may be substituted for the conventional parametric statistic.

In some cases, alternative statistics may be equivalent, in that they necessarily yield identical p values in a randomization test. For some designs, however, there may be two or more candidate statistics to be considered that are not equivalent. This is most likely to arise with designs, such as the single-case AB design, for which there is no analogous parametric test. In such cases, statistics based on t or F may not be equivalent to alternative statistics such as the difference between means (E.S.Edgington, personal communication, April 16, 1997). Then the statistic should be chosen that best reflects the expected treatment effect. Generally, this depends on whether or not there are expectations concerning differential

variability of control and treatment scores, a difference between their means, or both. The general issue of the interpretation of tests when treatments have different distributional effects as well as different effects on means was discussed in chapter 3. Once a statistic has been chosen, any alternative statistic that is equivalent in the sense just described may be substituted on grounds of computational simplicity (Edgington, 1995).

Computation of the Test Statistic for Arrangements of the Data

Systematic Arrangements of the Data.

One option is to compute the statistic for all possible arrangements of the observed data; that is, for every possible way that treatments could have been assigned to treatment occasions or participants, given the randomization procedure used. The proportion of arrangements yielding computed values for the statistic that are as large or larger than the obtained value is the probability that the obtained value occurred by chance. For some small group designs, there exist standard nonparametric rank tests, such as the Mann-Whitney test for inde-pendent groups and the Wilcoxon test for matched pairs. For these designs, provided there are enough participants to enter published tables of critical values, it is permissible to use the standard nonparametric tests. Similarly, for any single-case time-series design that is analogous to a group design (e.g., treatments are randomly assigned to times rather than to participants), the standard nonparametric rank test can be used. Due to loss of information as scores are converted to ranks, however, these are, at best, to be regarded as approximations to the randomization tests on which they are based. The statistical tables are in fact based on rearrangements of ranks, with no tied ranks, so they are only approximately valid when there are tied ranks in the data (Edgington, 1995).

Random Samples of Arrangements of the Data.

Instead of systematically computing the test statistic for all possible arrangements of the data, it is possible to use random sampling (with replacement) from all possible arrangements. For many designs, the total number of possible data arrangements is prohibitively large, even with the computational power of modern computers. For example, if the number of cups in the lady tasting tea experiment were to be doubled, the number of possible arrangements would increase from 70 to more than 600 million! Random sampling from all possible arrangements is clearly a sensible option in such cases. The usefulness of the random sampling approach is not limited, however, to designs with a very large number of possible arrangements, as it is sometimes awkward to list the possible arrangements even when the number of them is not enormous. Unfortunately, the random sampling approach has sometimes been referred to as providing an approximate test, in contrast to the exact test provided by systematic generation of all possible arrangements. This is misleading because a randomization test based on random sampling from arrangements of the data provides a fully valid p value (Edgington, 1969; Manly, 1991). The only limitation is that the power of such a test depends on the size of the random sample of arrangements. The larger the sample of data arrangements used, the greater the power of the test to detect an effect that would be significant using all possible arrangements.

The obtained data are treated (under the null hypothesis) as the first random sample from all possible arrangements and, assuming that a sample size of approximately 1000 is required, another 1000 arrangements are randomly generated. The probability of obtaining our actual data when the null hypothesis is true is given by the proportion of the 1001 test statistic values that are as large as the obtained value.

In the interests of consistency, we use random samples of possible arrangements of the data for all of our randomization tests. The question of how many samples are required is a matter of compromise between the desirability of having as high a power as possible and the extra time taken for the computation as more samples are added. How much of a problem the time factor is depends on the efficiency of the program that does the computation. In our case, we have implemented programs (macros) in three widely available commercial packages (Minitab, Excel, and SPSS). It is true that more efficient programs are possible in lower level languages (Onghena & May, 1995), but we think this misses the point. For researchers who are not computing enthusiasts, we believe that it is not the time costs that prevent them from using randomization tests; familiarity of the computing environment is probably more important. Of the three packages, Excel is probably the most widely available, but it is also the least efficient (slowest) for carrying out the necessary operations. For this reason, we have adopted sample sizes of 1000 for our implementation in Excel and 2000, for our implementations in Minitab and SPSS. However, it is easy for users to modify the macros to use 2000 (when working in Excel) or any other number of samples (when working in any of the packages) once they have ascertained (with the default sample size) that the time cost is likely to be acceptable. We provide guidelines, indicating how long each of our example computations took using each of the three packages on our computer, but considerable variations may be expected depending on details of users' computing environments. In any case, a sample of 1000 arrangements will often provide adequate power. Edgington (1969) used an illustrative example to argue that, for many purposes, power is adequate with a sample size of 1000, and Manly's (1991) tables for a range of sample sizes suggest that a sample size of 2000 should be adequate for most other cases.

Reporting the Statistical Decision

To avoid confusion with statistical decisions where interpretation of the statistic is based on conventional statistical tables, it is advisable to present the statistical conclusion for a randomization test in a way that makes clear how the conclusion has been reached. An example of how a statistical conclusion might be reported is now provided for the hypothetical study for which an outcome summary is presented in Table 4.3. Assuming a directional prediction, the statistical conclusion could be reported as follows:

> In a randomization test of the prediction that the child would maintain on-task behavior for longer when reinforcement was delivered by an age peer than when it was delivered by an adult, the proportion of data divisions giving a time difference in the predicted direction at least as large as the experimentally obtained difference was 0.029. Therefore, the obtained difference in on-task time between age peer and adult reinforcement periods was statistically significant ($p<0.05$; one-tailed).

This example can readily be adapted for reporting statistical conclusions for a randomization test applied to data from any other experimental design.

SUMMARY OF RANDOMIZATION TEST REQUIREMENTS

1. A randomization procedure is introduced into the experimental design.
2. A test statistic is selected and computed for the experimental data division. Where non-equivalent alternatives exist, the one that conforms most closely to expected effects is chosen. Where there are formally equivalent alternatives, the most convenient is chosen.
3. The test statistic is computed for alternative arrangements of the data that could have arisen from the randomization procedure incorporated in the design. The arrangements may be either systematic and exhaustive, or randomly sampled from all possible arrangements of the data. In the latter case, power to detect an effect increases with the size of the sample. Generally, 1000 to 2000 arrangements provide adequate power.
4. The proportion of data divisions giving a test statistic value at least as large as the experimentally obtained statistic indicates the significance level. This is reported in a way that makes clear that the statistical conclusion is based on reorderings of the data rather than on conventional statistical tables.

Chapter 5
*Randomization Designs for Single-Case and Small-*n *Studies*

A good case can be made for using randomization tests in large group studies when the assumptions of a conventional test are not met and the robustness of the test is in doubt (Chen & Dunlap, 1993). It is in the realm of single-case and small-*n* studies, however, that randomization tests have the greatest potential to increase confidence in the causal inferences made. In this chapter we provide examples of a range of standard single-case and small-*n* designs. For each one, we show how a random assignment procedure can be incorporated in the design. In chapter 2 we explained why we conclude that, even when the intention is to rely exclusively on visual analysis, incorporation of randomization procedures puts the causal inferences that we wish to make from our data on a more realistic footing.

For each design we provide an example of a research question to which it might be applied, together with illustrative data and other values, such as number of observations, that are necessary for a randomization test to be carried out. The illustrative data are presented as they would appear in a worksheet in preparation for analysis using one of our randomization test programs. The worksheet layout is identical for all three packages (Minitab, Excel, and SPSS) except that in Excel the data begin in Row 2 of the worksheet because the column labels occupy Row 1. Each worksheet is accompanied by an explanation of how the data are organized within it.

Macros (programs for doing the randomization tests) for use within each of the three packages, together with the illustrative worksheets, are available on the companion CD-ROM. In the following three chapters (one for each package), we explain first how the macros and the illustrative worksheets can be accessed, stored and, in the case of the worksheets, edited in preparation for analysis of the reader's own data. We also explain how to run the analyses within the chosen package. We then list the macros used to carry out randomization tests for each design, together with notes indicating what successive sections of the macros are doing. Each macro is followed by the results of an analysis of the illustrative data (including the time taken to run the analysis) and, where appropriate (i.e., for Minitab and Excel), a description of where the results will be located in the worksheet when the randomization test has been run.

The designs for which we provide examples broadly coincide with those we described in a paper directed at an audience of researchers in the area of aided communication (Todman & Dugard, 1999). We make use of some of the same examples here, but it should be obvious that examples pertinent to other research areas would be equally applicable. The designs are summarized in Table 5.1.

In Table 5.1, each design number is associated with a macro that will carry out a randomization test on data from that design. Where there are analogous small-*n* and single-case designs that can be analyzed using the same macro, they are given the same design number but each design is described separately. There are two instances (Designs 5a and 6a) where a different macro is provided for the special case of another design, in which only two treatments are applied and a directional prediction is possible.

DESIGN 1 (AB)

This is a single-case, two-phase design, usually with a control (i.e., baseline) condition followed by a treatment condition. A randomization test can be used to evaluate the difference between mean scores on a behavioral measure taken in the baseline and treatment phases, provided that the design is modified so that the point of intervention is randomly determined within a preset range of observation periods. It may be noted that, although the randomization test is valid, interpretation of the statistical conclusion is always compromised in this design by threats to internal validity, such as those arising from history and maturation effects. However, random determination of treatment onset may reduce the plausibility of such threats.

TABLE 5.1 Description of Designs

No.		*Design Title*		*Design Description*
1		AB		Single participant; 2 phases (baseline, treatment)
2		ABA Reversal		Single participant; 3 phases (baseline, treatment, return to baseline)
3		AB Multiple Baseline	or	Multiple participants; 2 phases (baseline, treatment) Single participant; multiple behavior; 2 phases (baseline, treatment)
4		ABA Multiple Baseline	or	Multiple participants; 3 phases (baseline, treatment, return to baseline) Single participant; multiple behavior; 3 phases (baseline, treatment, return to baseline)
5	and	One-Way Small-Groups Single-Case Randomized Treatment		2 or more treatments, equal or unequal group sizes Single participant; 2 or more treatments; equal or unequal numbers of occasions
5a		Small-Groups or Single-Case—2 Randomized Treatments	or	Special case of Design 5 with 2 treatments (directional prediction possible) Simple effects in a 2×2 single-case factorial (i.e., Design 7)
6	and	One-Way Small Group Repeated Measures Single-Case Randomized Blocks		3 or more treatments applied to 2 or more participants Single participant, with 3 or more treatments applied to 2 or more blocks

6a	Two Repeated Measures on Small Group or Single-Case Blocks	Special case of Design 6 with 2 treatments applied to each participant or each block (directional prediction possible)
7	Two-Way Factorial Single-Case	Single participant; 2 levels of each of 2 factors; equal numbers of observations at all combinations of levels
8	Ordinal Predictions	1 or more participants; 2 or more conditions

Example for Design 1

As an example, suppose that we want to know whether a user's rate of text entry into a communication aid is faster when a word prediction system is introduced. First, it will be necessary to decide on the total number of observation periods and the minimum number of periods for each phase. Suppose that 36 randomly ordered sentences of equivalent length and difficulty are to be entered and that there will be at least 8 in each of the two (control and treatment) phases. That means there will be 21 points at which the intervention could be introduced (i.e., anywhere from Sentence 9 to Sentence 29). The researcher must then randomly select one of these potential intervention points (i.e., as in a raffle procedure).

Design 1 Worksheet Box		
limits	**data**	**phase**
36	4	0
8	5	0
8	4	0
	3	0
	6	0
	4	0
	3	0
	4	0
	5	0
	3	0
	3	0
	4	0
	4	0
	5	0
	7	1
	8	1
	6	1

5	1
5	1
6	1
7	1
7	1
7	1
8	1
6	1
8	1
9	1
6	1
6	1
7	1
5	1
8	1
6	1
6	1
7	1
8	1

Suppose that observation period 15 was randomly selected as the intervention point and data were collected. The worksheet (with example data entered) would be as in the Design 1 Worksheet Box. Column 1 gives the total number of observations, the minimum number before intervention, and the minimum number from intervention. Observations are recorded in Column 2. Column 3 contains a zero against each observation before intervention and a one for every observation from intervention onward. Columns 1, 2, and 3 are named *limits, data,* and *phase,* respectively. When the data on rate of entry have been collected, a suitable statistic is computed. For this design, parametric statistics such as F or t are not equivalent (i.e., for the purpose of determining significance by data arrangements) to the difference between control and treatment means. The difference between control and treatment means seems likely to provide the best match with the treatment effect expected by a researcher and is therefore the one that is computed. For a nondirectional research hypothesis, the sign of the difference (positive or negative) is ignored to provide a two-tailed test statistic. If, however, the direction of the difference has been predicted, a one-tailed test statistic is obtained by using the difference between control and treatment means in the predicted direction. It should be noted that our one-tailed randomization test assumes that the intervention is expected to increase the mean score over phases (i.e., treatment mean>control mean) if a directional prediction is made. If this is not the case (i.e., the intervention is expected to reduce the mean score over phases), the easiest way to set up the worksheet is to transform the data by subtracting all observations from a convenient

number larger than any of them and to enter the transformed values in the data column. In the example here, as all scores are less than 10, this would be a suitable number from which to subtract each score.

In this example, if the first sentence to be entered using the word prediction system was Sentence 15, the two-tailed statistic would be the absolute difference (i.e., ignoring the direction of the difference) between the means for Sentences 1 to 14 and Sentences 15 to 36. Next, the same statistic is computed for 2000 (1000, in the case of Excel) randomly chosen intervention points that conform to the limits imposed by the experiment. This means sampling with replacement from the 21 possible assignments of the intervention point and computing differences between sentences in the two phases determined by the sampled intervention point. For example, when Sentence 9 is selected as the intervention point, the difference between Sentences 1 to 8 and 9 to 36 is computed, and when Sentence 10 is selected as the intervention point, the difference between Sentences 1 to 9 and 10 to 36 is computed, and so on. The null hypothesis is that rate of entry for each sentence is the same as it would have been if the other condition (control or treatment) had been given in that observation period. For the two-tailed test, the probability of getting a result at least as large as ours in either direction if the null hypothesis is true is given by the proportion of possible assignments of the intervention point for which the statistic is as large as or larger than the statistic obtained using the actual intervention point. In this example, if the largest absolute value (two-tailed) or the largest positive value (one-tailed) of the statistic is the one that uses the actual intervention point to split the observation periods into control and treatment phases, the probability of getting a result as extreme as ours if the null hypothesis is true is, subject to random sampling fluctuations, given by p=1/21=0.048. It is apparent that to stand a chance of obtaining a p value as low as 0.05 with this design, there will need to be at least 20 potential intervention points (i.e., p=1/20=0.05).

Although it must be conceded that the usefulness of this design is limited by threats to internal validity and lack of sensitivity, use of a randomization test for data analysis improves the situation somewhat. Differential (asymmetric) carryover effects present the usual interpretative problems for the randomization test but, unlike tests based on parametric significance tables, its validity is not compromised when nondifferential carryover (e.g., generalized practice) effects exist (Edgington, 1995). Furthermore, the design can be made more sensitive (as in Design 2) by adding a reversal (back to baseline) phase, provided this poses no ethical problems, or by addition of participants (or target behaviors) to create a multiple baseline design (as in Design 3).

DESIGN 2 (ABA REVERSAL)

This is a single-case, three-phase (usually baseline, followed by treatment, followed by withdrawal) design. Frequently, there are powerful ethical considerations that preclude the use of a reversal phase in clinical research. There are occasions, however, as in the following example, when the design may be acceptable. Provided that the design is modified so that the points at which treatment is introduced and withdrawn are randomly determined within a preset range, the difference between scores on a behavioral measure taken in the control (baseline plus reversal) phases and the treatment phase can be evaluated using a randomization test.

Design 2 Worksheet Box		
limits	**data**	**phase**
36	4	0
8	5	0
8	4	0
8	3	0
	6	0
	4	0
	3	0
	4	0
	5	0
	3	0
	3	0
	6	1
	5	1
	5	1
	7	1
	8	1
	6	1
	5	1
	7	1
	6	1
	7	1
	7	1
	7	1
	8	1
	4	0
	3	0
	5	0
	3	0
	5	0
	4	0
	3	0
	4	0
	4	0
	3	0
	4	0
	3	0

Example for Design 2

Consider an extension of the preceding AB design to test the hypothesis that neither the introduction nor withdrawal of the treatment has any effect. We need to decide on the total number of observation periods and the minimum number of periods for each phase. Suppose that, as before, 36 randomly ordered sentences are to be entered. We also stipulate that there will be at least 8 sentences in each of the three (baseline, treatment, and withdrawal) phases. This means, for example, that Point 9 can be paired with any of Points 17 to 29 (13 possible pairings), Point 10 can be paired with any of Points 18 to 29 (12 possible pairings), and so on, up to the pairing of Point 21 with Point 29. So we have 13+12+11+10+9+8+7+6+5+4+3+2+1=91 possible pairings of intervention and withdrawal points that meet this requirement. The researcher must then randomly select one of these pairs of potential intervention and withdrawal points. A convenient way of doing this would be to list systematically all possible pairs (i.e., 9–17, 9–18, 9–19, … 20–28, 20–29, 21–29) and number them 1 to 91 before randomly selecting a number in that range. Column 1 in the worksheet (displayed in the Design 2 Worksheet Box) gives the total number of observations, the minimum number before intervention, the minimum number during the intervention phase, and the minimum number in the withdrawal phase. Observations are recorded in Column 2. Column 3 contains a zero against each observation before intervention and a one for every observation from intervention until withdrawal, then zeros again until the end. Columns 1, 2, and 3 are again labeled *limits, data,* and *phase,* respectively. Again, a convenient statistic is the difference between control (i.e., preintervention plus postwithdrawal) and treatment means, with the difference in the predicted direction being used for a one-tailed statistic or the absolute difference for a two-tailed statistic. As for Design 1, our one-tailed randomization test assumes that the intervention is expected to increase mean scores if a directional prediction is made and, if this is not the case, all scores should be subtracted from a value that is greater than any of them.

The simulated data in the Design 2 Worksheet Box show the selected intervention and withdrawal points to be at observation periods 12 and 25. The two-tailed statistic would therefore be the absolute difference between the means for Sentences 1 to 11 plus 25 to 36 and Sentences 12 to 24. When the statistic has been computed using the actual intervention and withdrawal points, the same statistic is then computed for 2000 (1000 for Excel) randomly chosen pairs of intervention and withdrawal points that conform to the limits imposed by the experiment. This means sampling with replacement from the 91 possible pairs of intervention and withdrawal points. For the two-tailed test, the probability of getting a result as extreme as ours in either direction if the null hypothesis is true is given by the proportion of possible assignments of the pairs of intervention and withdrawal points for which the statistic is as large as or larger than the statistic obtained using the actual intervention and withdrawal points. In this example, the lowest possible p value, subject to random sampling fluctuations, would be 1/91=0.011. Thus, the addition of a withdrawal phase substantially lowers the p value achievable with a given number of observation periods. It should be noted, however, that rejection of the null hypothesis does not necessarily imply an effect of the treatment intervention. It could be an effect of both intervention and withdrawal, or even withdrawal alone. This should be reflected in any report of the statistical conclusion, for example, by beginning the report as follows: “In a randomization test of

the prediction that mean rate of entry would be faster when a word predic-tion system was used than in control phases before its introduction and after its withdrawal, the proportion of data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.044", where this is the one-tailed probability obtained when the randomization test is run.

DESIGN 3 (AB MULTIPLE BASELINE)

This is a multiple participant (or multiple behavior), two-phase (usu ally baseline, followed by treatment) design. The effect of treatment can be evaluated with a randomization test, provided that the design is modified so that the intervention points are randomly determined within a preset range for each participant (or for each behavior).

Design 3 Worksheet Box			
limits	**data**	**phase**	**part.**
3	4	0	1
18	5	0	1
6	4	0	1
6	3	0	1
	6	0	1
	4	0	1
	3	0	1
	4	0	1
	7	1	1
	8	1	1
	6	1	1
	6	1	1
	9	1	1
	8	1	1
	7	1	1
	7	1	1
	7	1	1
	8	1	1
	5	0	2
	5	0	2
	7	0	2
	8	0	2
	6	0	2
	5	0	2
	7	0	2
	6	0	2
	7	0	2
	7	0	2
	7	0	2
	8	1	2
	7	1	2

Continued next page

Continuation of Design 3 Worksheet Box			
limits	**right**	**phase**	**part.**
	6	1	2
	7	1	2
	7	1	2
	7	1	2
	8	1	2
	4	0	3
	3	0	3
	5	0	3
	3	0	3
	5	0	3
	4	0	3
	3	0	3
	6	1	3
	5	1	3
	4	1	3
	6	1	3
	6	1	3
	7	1	3
	6	1	3
	5	1	3
	6	1	3
	7	1	3
	6	1	3

Example for Design 3

Consider an extension of our example of a single-case AB design, in which we replicate the design with several different participants. As before, we are interested in the effect of introducing a word prediction system for text entry. From a statistical point of view, the advantage of multiple baselines is that effects can be detected with a smaller number of observation periods for each participant. For example, with three baselines, only three possible intervention points for each participant would be needed to make $p<0.05$ a possibility. Provided the intervention point is selected randomly for each baseline, there would be $3^3=27$ possible assignments, making a p value of $1/27=0.037$ a possibility in a randomization test. This contrasts markedly with the 20 possible intervention points required for the possibility of significance at $p<0.05$ in a single baseline AB design and makes the design particularly attractive when a long sequence of observations is impractical. It is a requirement

of this design that all participants (or behaviors) have the same total number of observation periods. Suppose that three suitable participants are available and that each can reasonably be given a total of 18 observation periods. We will stipulate that, for each participant, there must be at least six sessions in each phase, so the intervention can occur anywhere from Period 7 to Period 13. That is, the researcher must select randomly from 7 possible intervention points independently for each participant (i.e., the possibility of participants sharing the same intervention point is not excluded).

Column 1 in the worksheet (see Design 3 Worksheet Box) contains the number of participants, the number of observations per participant, the minimum before intervention, and the minimum after intervention. Column 2 contains the observations, Participant 1 followed by Participant 2 and so on until all participants are entered. Column 3 contains a zero opposite each observation before intervention and a one opposite each observation after intervention. Column 4 contains the participant number, starting at 1, for each observation. These columns are named *limits, data, phase,* and *participant,* but it should be borne in mind that the participant label may stand for behavior if the multiple baselines are for different behaviors rather than different participants. A convenient statistic will be the sum of the three differences between treatment and control means for the individual participants. As usual, the difference in the predicted direction is used for a one-tailed statistic or the absolute difference for a two-tailed statistic. As for the preceding designs, our one-tailed randomization test assumes that the intervention is expected to increase mean scores if a directional prediction is made.

In the simulated data in the Design 3 Worksheet Box, we can see that the randomly selected intervention points for the three participants were observation points 9, 12 and 8, respectively. The two-tailed statistic would therefore be the sum of the absolute differences between the means for Sentences 1 to 8 and 9 to 18 (Participant 1), 1 to 11 and 12 to 18 (Participant 2), and 1 to 7 and 8 to 18 (Participant 3). When the statistic has been computed using the actual intervention points, the same statistic is then computed for 2000 (1000 for Excel) randomly chosen intervention points for the three participants that conform to the limits imposed by the experiment. In this example, there are 7^3=343 possible arrangements of the data to be sampled with replacement, so the lowest possible p value, subject to random sampling fluctuations, would be 1/343=0.003. Once again, it is apparent that the sensitivity of the randomization test is greatly increased by the modification to Design 1.

An example of an AB multiple baseline design with replications across behaviors is now given. Suppose we wished to investigate the effects of training a single communication aid user in the use of three conversational techniques. In addition to hypothesizing that the frequency of use of these techniques will actually increase with training, we might hypothesize that increased use of the techniques will affect the impression ratings given by conversational partners. For example, we might hypothesize that training in the use of "turn-around" questions, idiosyncratic comments, and feedback remarks will lead to the user being perceived as more competent, interesting, and interested, respectively. As in the preceding example, three intervention points would be selected randomly, in this case from the sessions planned for a single participant, thus providing each of the behavioral measures with its own control and treatment phases. As the AB multiple baseline design with replications across participants is computationally identical, from the point of view of a randomization test, to the case with replications across behaviors, we do not provide a numerical example for the latter.

Although the randomization test specified for the AB multiple baseline design is completely valid, some care is needed in the interpretation of the obtained *p* value (Edgington, 1996). Briefly, a significant result does not allow any inference about which one or more participants were affected, or which one or more treatments affected which one or more behavioral measures. It may nonetheless sometimes be quite helpful to be confident that the treatment was effective for at least one participant, or that at least one of the treatments had an effect on at least one of the behavioral measures. Such information may well provide the necessary assurance that the research issue is worth pursuing in further studies designed to tease apart the controlling factors that are confounded in the initial study. Any report of the statistical conclusion for a multiple baseline design should make it clear that participants or behaviors are not differentiated with respect to any statistically significant effect that has been found, although visual inspection may be expected to provide some indication of where the effect is likely to have occurred.

DESIGN 4 (ABA MULTIPLE BASELINE)

This is a.multiple participant (or multiple behavior), three-phase (usually baseline, followed by treatment, followed by withdrawal) design. The effect of treatment can be evaluated with a randomization test provided that the points at which treatment is introduced and withdrawn are randomly determined within a preset range for each participant (or for each behavior). This design combines the advantages of reversal and multiple baseline designs to make it possible to reveal significant effects with small numbers of observation sessions combined with small numbers of replications over participants or behaviors. The same ethical considerations as for Design 2 (ABA Reversal) apply.

Example for Design 4

As the randomization test is identical for data generated from a multiple participant and a multiple behavior design, and the multiple behavior example from Design 3 is readily adaptable to this design, we confine ourselves to an example of a multiple participant design. Considering the same general hypothesis about the effect of a word prediction system on rate of text entry as for the preceding designs, suppose that we have only two participants available and that only 12 observation sessions per participant are practical. We stipulate that, for each participant, there will be at least three sentences in each of the three (baseline, treatment, and withdrawal) phases. This gives 10 possible pairs of intervention and withdrawal points for each participant, so that $p=1/10^2=0.01$ is a possibility if the observed value of the statistic (the difference between control and treatment means) is as high as or higher than the values for all other possible arrangements of the data. The researcher must randomly select one of the possible pairs of intervention and withdrawal points for each participant. As for Design 2, a convenient way of doing this would be to list systematically all possible pairs (i.e., 4–7, 4–8, 4–9, 4–10, 5–8, 5–9, 5–10, 6–9, 6–10, 7–10) and number them 1 to 10 before randomly selecting a number in that range for each participant.

Design 4 Worksheet Box			
limits	**data**	**phase**	**part.**
2	4	0	1
12	5	0	1
3	4	0	1
3	3	0	1
3	5	0	1
	7	1	1
	6	1	1
	5	1	1
	7	1	1
	5	0	1
	6	0	1
	4	0	1
	6	0	2
	5	0	2
	7	0	2
	8	1	2
	9	1	2
	9	1	2
	8	1	2
	6	0	2
	8	0	2
	7	0	2
	7	0	2
	6	0	2

As in Design 3, all participants must have the same total number of observations. Column 1 in the worksheet (see Design 4 Worksheet Box) contains the number of participants, the number of observations per participant, the minimum number before intervention, the minimum number during the intervention phase, and the minimum number in the withdrawal phase. Column 2 contains the observations, Participant 1 followed by Participant 2, and so on until all participants are entered. Column 3 contains a zero opposite each observation before the intervention and from withdrawal to the end and a one opposite each observation during the intervention phase. Column 4 contains the participant number, starting at 1, for each observation. These columns are again named *limits, data, phase,* and *participant.* As for Design 3, however, the *participant* label may stand for behavior if the multiple baselines are for different behaviors rather than different participants. The test statistic is the sum over participants of the difference between the mean of observations taken during the

intervention phase and the mean of observations taken before intervention and after withdrawal. As for the preceding three designs, our one-tailed randomization test assumes that the intervention is expected to increase mean scores if a directional prediction is made.

We see from the simulated data in the Design 4 Worksheet Box that the randomly selected pairs of intervention and withdrawal points for the two participants were Points 6 and 10 and 4 and 8 respectively. The two-tailed statistic would therefore be the sum of the absolute differences between the means for Sentences 1 to 5 plus 10 to 12 and 6 to 9 (Participant 1) and between Sentences 1 to 3 plus 8 to 12 and 4 to 7 (Participant 2). When the statistic has been computed using the actual intervention points, the same statistic is then computed for 2000 (1000 for Excel) randomly chosen intervention and withdrawal pairs for the two participants that conform to the limits imposed by the experiment.

Both of the interpretative limitations that were noted for Design 2 (inability to separate the effects of intervention and withdrawal) and Design 3 (inability to separate the effects on different participants or behaviors) apply to this design and should be acknowledged in any report of the statistical conclusion.

DESIGN 5

This design is really two in one. There are two designs, one for small groups and one for a single participant, that are precisely equivalent in terms of the randomization test required. However, as these two de-signs seem quite distinct conceptually, we provide a separate example for each. It may be noted that this design is essentially the same as that used in Fisher's lady tasting tea example, except that the observations are scores rather than binary decisions. This design, together with Design 6, is sometimes described as an *alternating design,* to distinguish it from phase designs such as Designs 1 through 4.

Design 5 (One-Way Small Groups)

For this design (the same applies to the small group design version of Design 6) the logic of the design does not differ from that of a corresponding large group design. It is only the appropriate statistical analysis that differs. In this design there is one manipulated variable, with random assignment of participants to treatment groups. Group sizes may be equal or unequal. For this design, as for all of the designs included within Designs 5 and 6, it would be acceptable to use a standard nonparametric alternative to a randomization test, in this case the Kruskal-Wallis test. However, this is not the preferred solution because, as discussed in the section on parametric tests in chapter 1, information would be lost unnecessarily in the conversion of scores to ranks and, in addition, the tests are only approximate when there are tied ranks. The randomization test is analogous to a between subjects one-way ANOVA, which would often be the test of choice in an equivalent large-*n* design. With just six participants divided equally between two groups, a randomization test makes it possible (with systematic generation of all possible data arrangements) to obtain a *p* value of 0.05 and with six participants divided equally between three groups the smallest possible *p* value would be 0.011.

Example for Design 5 (small groups).

Suppose that we have 10 participants willing to try any one of four communication aids and we want to know whether the systems differ in the time that it takes to learn to use them to some criterion. Suppose also that we have available two each of two of the systems and three each of the other two systems. With participants randomly assigned to the communication aids, the total number of possible arrangements is given by 10!/(3!3!2!2!)=25200. Column 1 in the worksheet (see Design 5 Worksheet Box) gives the total number of observations

Design 5 Worksheet Box		
limits	**data**	**condit.**
10	4	1
	4	1
	2	2
	4	2
	5	3
	4	3
	5	3
	7	4
	6	4
	6	4

Simulated observations are recorded in Column 2, and Column 3 contains the treatment group codes starting at 1. As is apparent in the example, numbers in the groups need not be equal. The columns are labeled *limits, data,* and *condition.* For this design, *F* is an appropriate statistic, but residual sum of squares (RSS) is equivalent to F for the purpose of the randomization test and is the statistic used because it is simpler to compute. When the statistic (RSS) has been computed for the actual data, it is then computed for 2000 (1000 for Excel) randomly chosen arrangements of the data that conform to the numbers of observations per condition used in the experiment. The proportion of RSS values that are smaller than the actual RSS value is the required probability. It should be noted that, for most statistics, it is the proportion of rearrangement values that are higher than the actual value that gives the probability. The reverse is true in the case of the RSS statistic because it is the residual (error) sum of squares. As there are 25200 possible arrangements to be sampled with replacement, the lowest possible *p* value, subject to random sampling fluctuations, would be 1/25200= 0.00004.

Design 5 (Single-Case Randomized Treatment)

This is like the preceding design except that treatments are randomly assigned to treatment occasions for one participant, rather than being randomly assigned to different participants.

The randomization test is identical to that for the one-way small group design, so the same worksheet (Design 5 Worksheet Box), with the same simulated data, is used to illustrate an example of the single-case randomized treatment design.

Example for Design 5 (single-case).

Suppose that we are interested in the effect of translucency of graphic symbols on the number of sessions required by a communication aid user to learn the symbol-referent associations of sets of 20 symbols. We assume that translucency ratings for some suitable symbols are available and that when they are classified as belonging to one of four translucency ranges (high, medium/high, medium/low, low), the number of sets of 20 symbols that can be constructed in each range is two for each of the high and medium/high ranges and three for each of the low and medium/low ranges. The 10 sets of symbols should be randomly assigned to 10 available periods. The simulated data are as displayed in the Design 5 Worksheet Box and all details of the computation of the test statistic are the same as for the previous small group design.

Design 5a (Small Groups Or Single-Case—2 Randomized Treatments)

This is a special case of Design 5, in which the number of levels of the manipulated variable is restricted to two. It is given separately because, unlike Design 5, a directional prediction maybe specified. It is also convenient for testing simple effects in a two-by-two factorial design (see Design 7). In this design there is one manipulated variable with two levels, with random assignment of participants to treatment groups (for the small groups variant of the design) or random assignment of observation periods to treatment conditions (for the single-case variant of the design). Group sizes may again be equal or unequal.

Example for Design 5a.

Suppose that in the preceding examples for Design 5 we had only nine communication aids (four of one kind and five of another) or nine sets of translucency symbols (four high and five low translucency). In the translucency example, we might predict that high-translucency symbols will give lower means (i.e., will require fewer sessions to learn) than low-translucency symbols. The nine sets of symbols should be assigned randomly to nine available treatment periods for a single participant, making the lowest possible p value, subject to random sampling fluctuations, equal to $1/(9!/(5!4!))=0.0079$. Column 1 in the worksheet (see Design 5a Worksheet Box) contains the number of observations. The observations are in Column 2 and Column 3 contains the factor level codes, 1 and 2, opposite the observations. As in Design 5, the columns are named *limits, data,* and *condition*. The test statistic used is the difference between condition means (Condition 2–Condition 1). It should be noted that our one-tailed randomization test assumes that the prediction is that the mean for Condition 2 will be greater than that for Condition 1 (i.e., low translucency must be Condition 2 in our example). When the statistic has been computed for the actual

data, it is then computed for 2000 (1000 for Excel) randomly chosen arrangements of the data that conform to the numbers of observations per condition used in the experiment. For a one-tailed test, the probability will be given by the proportion of values of the directional

Design 5a Worksheet Box		
limits	**data**	**condit.**
9	4	1
	4	1
	5	1
	4	1
	4	2
	6	2
	7	2
	6	2
	6	2

statistic (Condition 2–Condition 1) for rearrangements of the data that are at least as large as the statistic for the actual data. For a two-tailed test, the absolute difference between the means is used.

DESIGN 6

As for Design 5, this design is really two in one. The two designs are for small groups and for a single participant. They are equivalent in terms of the randomization test required but, once again, as they seem conceptually distinct, we provide a separate example for each.

Design 6 (One-Way Small Group Repeated Measures)

In this design there is one manipulated variable, with repeated measures on conditions for each of several participants. Participants must receive the same number of measures, with the order of conditions randomized independently for each participant. For this design, the Friedman ANOVA would be an acceptable, although less satisfactory, nonparametric alternative. The randomization test is also analogous to a one-way ANOVA with repeated measures. With just three levels of the manipulated variable and two participants, it is possible to obtain a p value of 0.05 and with three levels of the variable and three participants a p value of 0.01 is possible.

Example for Design 6 (small group).

Suppose that we want to know about the effect of a conversational partner's experience on the number of topic initiations made by users of a communication aid. We designate four

levels of experience: no experience interacting with users, occasional experience of such interactions, regular experience of such interactions, and regular experience augmented by video training sessions. We assume that three users of communication aids are available and for each user we can provide four conversational partners, one at each level of experience. The order in which each user's four conversational partners interact with them is randomly determined and the number of each user's topic initiations is recorded for each interaction. Column 1 in the worksheet (see Design 6 Worksheet Box) contains the number of observations (this must be number of conditions×number of participants), and, below that, the number of participants. The observations are in Column 2. All observations from the first participant must be entered first, then all the observations from the second participant must be entered, and so on, until all the observations have been entered in Column 2. The treatment conditions must be coded numerically starting at 1, and they must be in Column 3. The participants must also be coded starting at 1, and must be in Column 4. The

Design 6 Worksheet Box

limits	**data**	**condit.**	**part/block**
12	20	1	1
3	34	2	1
	51	3	1
	48	4	1
	43	1	2
	57	2	2
	62	3	2
	60	4	2
	56	1	3
	74	2	3
	90	3	3
	86	4	3

columns are named *limits, data, condition,* and *participant/block*. The statistic used is RSS, and it is computed for the actual data and 2000 (1000 for Excel) randomly generated arrangements of treatments within participants. As for Design 5, to be statistically significant, the value of RSS for the observed data ordering must be among the lower values obtained, rather than among the higher values.

Design 6 (Single-Case Randomized Blocks)

This is like the preceding design except that repeated measures are randomly assigned to blocks of time for a single participant, rather than to different participants. Available treatment times are grouped into blocks of adjacent times to control for variability over the period during which the experiment is conducted. Treatment conditions are assigned

randomly within each block of times, independently for each block. As in the small groups variant of this design, the randomization test is analogous to a one-way repeated measures ANOVA, where treatment order has been randomized independently for each participant (or block of times, in this case). Three treatments repeated in two blocks of times would be sufficient to make a p value of 0.05 possible, and a p value of 0.01 would be possible if the number of blocks was increased to three.

Example for Design 6 (single-case).

Suppose we are interested in the amount of time a communication aid user, who has access to both a high-tech and a low-tech aid, interacts using the high-tech aid with different partners. We specify four partners: the user's key worker at a day center, the user's speech and language therapist, the user's best friend, and a family member. We also specify three blocks of 1 day each, with four 2-hr sessions (morning, early afternoon, late afternoon, and evening) in each block. The assignment of partners to morning, early afternoon, late afternoon, and evening sessions is randomly determined for each 1-day block and the time (in min) spent using the high-tech aid is recorded for each interaction period. The simulated data are as displayed in the Design 6 Worksheet Box and all details of the computation of the test statistic are the same as for the previous small group design. That is, the statistic used is RSS, and it is computed for the actual data and for a random sample of 2000 (1000 for Excel) rearrangements of the data. Again, to be statistically significant, the value of RSS for the observed data ordering must be among the lower values obtained, rather than among the higher values.

DESIGN 6a (2 REPEATED MEASURES OR 2 RANDOMIZED BLOCKS)

This is a special case of Design 6, in which the number of levels of the manipulated variable is restricted to two. It is given separately because, as for Design 5a, a directional prediction may be specified. In this design there are repeated measures on the two levels of the manipulated variable. The order of treatment conditions is randomized independently for each participant (for the small group variant of the design) or within each block of times (for the single-case blocks variant of the design).

Example for Design 6a.

Suppose that in the preceding examples for Design 6 we were concerned with the effect of only two levels of experience (small group example) or only two categories of conversational partner (single-case example). Following up the single-case example, suppose the two categories of conversational partner of interest were speech and language therapist and family member. Suppose also that seven blocks of 1 day each are available, with a morning and afternoon session in each block (the equivalent situation for the small groups variant would be that seven communication users were available). Next, we assume that the

research hypothesis in the single-case study is that the communication aid user will use the high-tech aid more with the speech and language therapist than with the family member. Column 1 in the worksheet (see Design 6a Worksheet Box) contains the number of participants (small group variant) or the number of blocks of time (single-case variant). Column 2 contains the observations, of which there must be twice as many as the number of

Design 6a Worksheet Box

limits	data	condit.	part./block
7	4	1	1
	5	2	1
	3	1	2
	6	2	2
	5	1	3
	7	2	3
	6	1	4
	5	2	4
	5	1	5
	5	2	5
	3	1	6
	8	2	6
	3	1	7
	6	2	7

participants. Column 3 contains the condition code (1 or 2) and Column 4 the participant or block numbers. The data must be listed one participant (or block) at a time; that is, Participant or Block 1, first and second observation, then Participant or Block 2, and so on. Each participant or block must have one observation for each of Conditions 1 and 2. If a directional prediction is being made, code 2 must refer to the condition for which the higher mean is predicted. In this single-case example, for instance, the speech and language therapist condition must be coded 2. As in Design 6, the columns are named *limits, data, condition,* and *participant/block*. The statistic used for the one-tailed test is (Condition 2 mean—Condition 1 mean) and this is computed for the actual data and 2000 (1000 for Excel) randomly generated arrangements of treatments within participants or treatments within blocks. The one-tailed *p* value is given by the proportion of arrangements with a statistic at least as large as the value for the actual data, and the two-tailed probability is the proportion of absolute values of arrangement statistics at least as large.

DESIGN 7 (TWO-WAY FACTORIAL SINGLE-CASE)

This is a factorial design with two levels of each of two factors, where the number of observations in each of the four conditions must be equal. Random assignment of the conditions to treatment times is required. The randomization test is analogous to a between-subjects two-way ANOVA for the main effects of the two factors. Unfortunately, however, there is

no valid randomization procedure for testing for an interaction effect in this design (Edgington, 1995). However, it is possible to test for the four simple effects (i.e., the effects of Factor 1 at Level 1 of Factor 2; Factor 1 at Level 2 of Factor 2; Factor 2 at Level 1 of Factor 1; Factor 2 at Level 2 of Factor 1). These are the tests that would normally be carried out after a significant interaction effect has been found to describe fully the nature of the obtained interaction. One- and two-tailed tests are available for all main effects and simple effects. Two sessions in each condition would be sufficient to make a p value of 0.05 possible for the main effects and three sessions in each condition would be sufficient to make a p value of 0.05 possible for a simple effect.

Example for Design 7

Suppose that we are interested in how the rate of communication of a communication aid user is affected by alternative interface devices (touch-screen or joystick) and alternative display modes (dynamic or static). Assume that we predict an interaction, such that communication rates will be similar for all conditions except for the combination of touch-screen input with a static screen display, which will result in a slower communication rate. We assume

Design 7 Worksheet Box			
limits	**data**	**factor1**	**factor2**
16	4	1	1
	5	1	1
	3	1	1
	5	1	1
	7	1	2
	8	1	2
	6	1	2
	9	1	2
	8	2	1
	6	2	1
	7	2	1
	7	2	1
	7	2	2
	7	2	2
	7	2	2
	6	2	2

that 16 treatment times are available and that 4 will be allocated to each of the touch-screen/dynamic, touch-screen/static, joystick/dynamic and joystick/static conditions, with the assignment of treatment times to condition sessions being random. Communication rate in words per minute is recorded during each of the treatment periods. Column 1 in the

worksheet (see Design 7 Worksheet Box) contains the number of observations in the top cell. The observations are in Column 2. The Factor 1 levels are in Column 3, and the Factor 2 levels are in Column 4. The columns are labeled *limits, data factor1, and factor2*. For both factors the levels must be coded 1 and 2, and if the one-tailed test is to be used, Level 2 should denote the one with the higher predicted mean. In our example, Factor 1 is interface device, with touch-screen coded 1 and joystick coded 2 because the communication rate with a touch-screen interface is predicted to be lower on average than with a joystick interface. Factor 2 is display mode, with static coded 1 and dynamic coded 2 because the communication rate with a static display is predicted to be lower on average than with a dynamic display. The rationale for the coding decisions should be apparent from inspection of Fig. 5.1, which shows the nature of the predicted interaction. The test statistic is the difference between means (Level 2 mean -Level 1 mean) and this is computed for the actual data and for 2000 (1000 for Excel) random arrangements of the data for each factor.

Design 5a Worksheet Box (Simple Effects Example)

limits	**data**	**condit.**
8	4	1
	5	1
	3	1
	5	1
	7	2
	8	2
	6	2
	9	2

Looking at the example data in the Design 7 Worksheet Box, we note that the totals in each cell are consistent with our predicted inter-action (i.e., touch-screen/static=17; joystick/static=28; touch-screen/dynamic=30; joystick/dynamic=27) and, consistent with this interaction, that if there are any main effects they would be in the direction dynamic>static and joystick>touch-screen. The main effects are tested using the macro for the present design and the simple effects of each factor are tested using the macro for the two randomized treatments analysis, which was presented as Design 5a. For example, suppose we wish to test the simple effect of display mode (dynamic [2] vs. static [1]) for the touch-screen interface. The worksheet for Design 5a would contain the top half of the data from the Design 7 Worksheet Box (i.e., the data for the touch-screen interface only) in the data column (Column 2). The condition column (Column 3) in the Design 5a Worksheet would contain

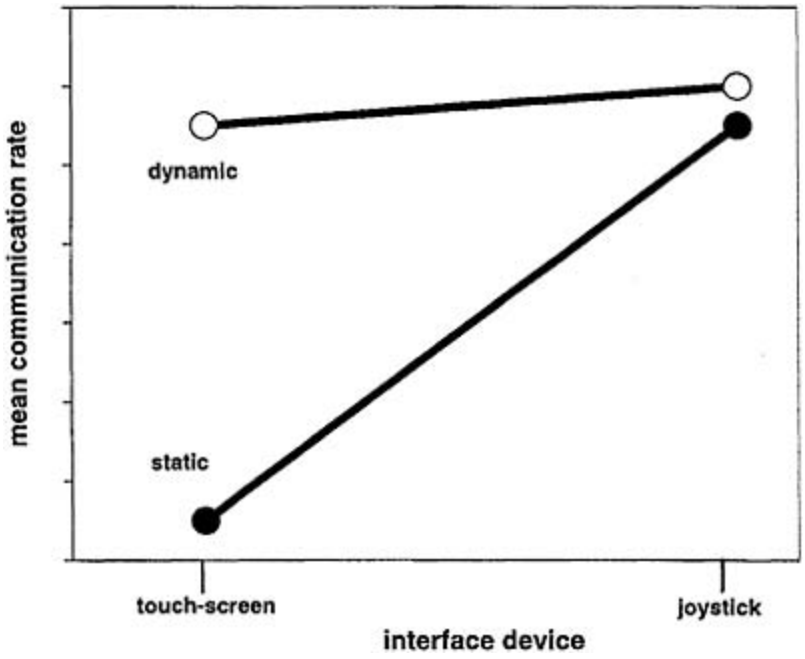

FIG. 5.1. Predicted communication rates.

the top half of the codes under Factor 2 from the Design 7 Worksheet Box (i.e., four 1s followed by four 2s), and the number of observations in the limits column (Column 1) would be 8. The entries in the Design 5a Worksheet to test this simple effect are shown in the Design 5a Worksheet Box (Simple Effects Example).

DESIGN 8 (ORDINAL PREDICTIONS)

There are designs in which clear ordinal predictions may be made and we end our list of designs with small-*n* and single-case examples of such a design. The randomization test that we present is analogous to a correlation between predicted and obtained orders for scores on a dependent variable.

Example for Design 8

Consider the suggestion that people who have been used to conversing before losing the ability to speak are likely to be more socially effective users of a conversation aid than people who have never been able to speak. Consider also the suggestion that physical disabilities may contribute to an impression of communicative incompetence. Now, suppose we have six users of a communication aid, three of whom have lost the ability to speak and three of whom have never been able to speak. Suppose we can rank the members of each

subgroup as high, medium, and low in terms of degree of physical disability. We might then hypothesize that there will be a unique ordering of our six communication aid users on ratings of communicative competence (i.e., from never had speech/high physical disability at the low end to lost speech/low physical disability at the high end). Even if a unique ordering is not possible, a partial ordering may still be worth considering. For example, a partial order for six participants could be based on the prediction that three who never had speech will be rated less competent than three who lost speech.

It is also possible to envisage occasions when ordinal predictions might be made for measures on a single participant. For example, suppose that a communication aid user has conversations with six partners. It might be predicted that the user will ask more questions of an opposite-sex partner than of a same-sex partner. It might also be predicted that the closer the partner is to the user's own age, the more questions the user will tend to ask. Then, if three each of the partners are male and female, and males and females can be ordered separately with respect to their age similarity to the user, a unique order can be predicted for the frequency of user questions. The form of the prediction is the same as that suggested for the small group study discussed earlier. Again, if only a partial order can be predicted on the basis of gender, the prediction is identical in form to the partial order suggested for the small group case.

Even though we are working with classification variables, it may be possible to assign individuals (users or conversational partners in the two examples) randomly to rating occasions. Their individual characteristics of interest (the independent variables) would, of course, also be randomly assigned, and a valid randomization test of our ordinal hypothesis would be possible. In the absence of a random assignment of units to observation periods, a randomization test would be statistically invalid. However, we take up the issue of whether it may ever be acceptable to use this randomization test without random assignment in a later chapter. For now, we proceed on the assumption that random assignment has been used. Communicative competence ratings of the six participants are obtained. In the Design 8 Worksheet Box (Unique Order Example), Column 1 contains the number of observations. The actual (simulated) observations are in Column 2 and the corresponding predicted unique order is in Column 3, with 1 for the observation(s) predicted to be lowest. If the ordering is only partial, the same value is assigned to observations for which an order cannot be predicted. An illustrative worksheet for a predicted partial order is provided in Design 8 Worksheet Box (Partial Order Example). The columns for both unique and partial

Design 8 Worksheet Box (Unique Order Example)

limits	**data**	**predict**
6	3	1
	2	2
	5	3
	4	4
	6	5
	7	6

predicted orderings are named *limits, data,* and *predict*. The sum of products of the data and the predicted order is the statistic used and this is computed for the actual data and for 2000 (1,000 for Excel) random rearrangements of the data. As indicated earlier, this is analogous to computing the correlation between predicted and obtained orders.

Design 8 Worksheet Box (Partial Order Example)		
limits	**data**	**predict**
6	3	1
	2	1
	5	1
	4	2
	6	2
	7	2

A unique ordering of five participants (or five measures on a single participant) would be sufficient to make possible a p value of 0.01, and even a partial ordering of six participants (or measures on a single participant) would be sufficient for a p value of 0.05.

Chapter 6
Randomization Tests Using Minitab

All of the designs described in chapter 5 can be subjected to randomization tests within Minitab for Windows using the same general procedures. We suggest that you begin by copying all of the files in the Minitab subdirectory on the companion CD-ROM to a convenient directory. There are two kinds of files there: Minitab macros with .txt extensions (*des1.txt* to *des8.txt*) and Minitab worksheets with .mtw extensions (*des1.mtw* to *des8unique.mtw*). The next step, once this has been done, is to edit the worksheet that corresponds to the design of interest so that it contains your own data, remaining in conformity with the worksheet specifications given in chapter 5. To edit one of the worksheets (e.g., des1.mtw to analyze Design 1 data) open it in Minitab and simply replace the numerical entries in all columns with your own values. Then, if you wish, save the current worksheet in the same directory, calling it owndes1 or something similar (the .mtw extension will be added automatically). To run the macro, with the worksheet open, type in the session window the command %, followed by the full path name to the macro, followed by the name of the macro. For example, if the macro des1 .txt had been stored in a subsubdirectory (called clinical) of a subdirectory (called analyses) of the C: drive, the complete command would be:

%c:\analyses\clincal\des1.txt

If the macro had been on a floppy disk in the A: drive containing no subdirectories, the complete command would be:

%a:\des1.txt

In the remainder of the chapter, the following are listed for each design:

1. The design specifications for the example in chapter 5.
2. The macro for the design, with bolded comments indicating the function of each section of the macro.
3. The line numbers of the macro in which the number of samples are specified (2000 or 2001, including the actual data) so that the user can easily locate these values to change them (e.g., to 5000 and 5001) if required.
4. The location of the randomization test results on the worksheet when the macro has been run.
5. The randomization test results for three runs of the macro on the example worksheet, including the average time taken for the run using Minitab for Windows on a Pentium P266 MHZ with 64 Mb RAM and an Intel processor with no other packages running concurrently (all of the designs took less than 5 min to run).
6. A statement of the statistical conclusion based on the first run of the macro (which normally would be the only result obtained).

It should be noted that the count of rearrangement statistics that are at least as large (or in some cases, at least as small) as the test statistic, and its probability under the null hypothesis, will vary slightly from run to run on the same data. This is due to random sampling fluctuations as 2000 arrangements are randomly sampled. Provision of the results of three runs of the macro should help users to gain some idea of the extent of variation using 2000 samples. We have generally provided sample data that would give probabilities close to a critical value (e.g., p =0.05) if systematic sampling of all possible arrangements were generated. For example, in Design 1, systematic generation of all 21 possible arrangements of the data would yield a probability of 1/21=0.048. Consequently, using random sampling of arrangements with replacement, the statistical decisions obtained sometimes differ between runs with the same data. If your own data produce probabilities close to a critical value, you may wish to obtain more stable probability estimates than those provided by 2000 samples. However, the propriety of sequential testing of this kind in the absence of taking any steps to protect the significance level is open to question and, ideally, a decision about how many samples to use should be taken at the outset (see the discussion of this issue in chap. 12). If you nonetheless decide to pursue a sequential strategy or you decide at the outset to use a different number of samples, you should open the macro in a convenient word processor (e.g., Microsoft Word for Windows), locate the 2000 and 2001 entries and change them to, say, 5000 and 5001. The macro should then be saved with the same name in text-only format in the same directory. This will, of course increase the length of the run time, so it may be a good idea to try a run, possibly using simulated data comparable to your real data, with the original 2000 rearrangements on your own computer to see how long that takes before making a change.

It should also be noted that in some of the designs, one- and two-tailed probabilities often, although not necessarily, will be identical. For example in the single-case AB design (Design 1), the probabilities will only differ when one or more arrangements of the data produces a difference between means in the nonpredicted direction that is at least as large as the observed difference between means.

DESIGN 1 (AB)

Specifications for Design 1 Example

Total number of observation periods	=36
Minimum number of baseline periods	=8
Minimum number of treatment periods	=8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention point for the example is at Period 15.

Commented Macro for Design 1 (Macro File Name: des1.txt)

```
gmacro
des1
let k1=c1 (1)                          collect information on number
                                       of observations, minimum
                                       number of preintervention
                                       periods, and minimum number
                                       of intervention periods from
                                       column 1
let k2=c1(2)
let k3=c1(3)
let k10=k2+1                           find the first and last possible
                                       intervention points
let k11=k1−k3+1                        number of arrangements
do k20=1:2000                          find a random intervention
random 1 c4;                           point for this arrangement
integer k10: k11.
let k14=c4(1)−1
let k15=k1−k14
set c5                                 make a column of zeros and
                                       ones for the random
                                       intervention point
k14(0) k15(1)                          find the difference between
end                                    means for this arrangement,
let c6=c2*c5                           then store
let c7(k20)=sum(c6)/k15−(sum(c2)−sum(c6))/k14
enddo                                  next arrangement
let c6=c2*c3                           now get the actual test statistic
let k30=sum(c6)/sum(c3)−(sum(c2)−sum(c6))/(k1−sum(c3))
let c8=c7−k30                          subtract from the arrangement
                                       test statistics and code zero or
                                       one
code (−1000000:−0.0000001)0 (0:1000000)1 c8 c9
let c10=abs(c7)−abs(k30)               same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c10 c11
let c12(1)=k30                         display actual test statistic at
                                       the top of column 12
let c12(2)=sum(c9)                     count the arrangement statistics
                                       that are at least as large
let c12(3)=c12(2)/2001                 and find the one-tailed
                                       probability
```

```
let c12(4)=abs(k30)          same for absolute values
let c12(5)=sum(c11)          and find the two-tailed
let c12(6)=c12(5)/2001       probability
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 8 for 2000 entry
Lines 2 and 5 up from bottom for 2001 entries

Location of Design 1 Test Results

After running, the test statistic (for the actual data) is at the top of Column 12 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 1 Example

Row	*Result*	*1st run c12*	*2nd run c12*	*3rd run c12*
1	One-tailed statistic	2.656	2.656	2.656
2	No. as large	83	82	100
3	One-tailed probability	0.042	0.041	0.050
4	Two-tailed statistic	2.656	2.656	2.656
5	No. as large	83	82	100
6	Two-tailed probability	0.042	0.041	0.050

Mean time for three runs=54 sec

Statistical Conclusion for Design 1 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would increase when a word prediction system was introduced, the proportion of 2000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.042. Therefore, the obtained difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 2 (ABA REVERSAL)

Specifications for Design 2 Example

Total number of observation periods	=	36
Minimum number of control periods before treatment starts	=	8

Minimum number of treatment periods	=	8
Minimum number of control periods after withdrawal of treatment	=	8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pair of intervention and withdrawal points for the example is at Periods 12 and 25.

Commented Macro for Design 2 (Macro File Name: des2.txt)

```
gmacro                                  collect information on number of
des2                                    observations, minimum number of
let k1=c1(1)                            preintervention periods, minimum
                                        number of intervention periods
                                        and number of withdrawal periods
                                        from column 1
let k2=c1(2)                            find the first and last possible
let k3=c1(3)                            intervention points
let k4=c1(4)
let k10=k2+1
let k11=k1−k3−k4+1                      number of arrangements
do k20=1:2000                           find a random intervention point for
random 1 c4;                            this arrangement
integer k10:k11.                        find the first and last possible
let k12=c4(1)+k3                        withdrawal points
let k13=k1−k4+1
if k12=k13                              find a random withdrawal point for
let c5(1)=k1−k4+1                       this arrangement
else
random 1 c5;
integer k12:k13.                        make a column of zeros and ones for
endif
let k14= c4(1)−1
                                        the random intervention and
                                        withdrawal points
let k15=c5(1)−c4(1)                     find the difference between
let k16=k1−c5(1)+1                      means for this arrangement, then
set c6                                  store
k14(0) k15(1) k16(0)
end
let c7=c2*c6
let c8(k20)=sum(c7)/k15−(sum(c2)−sum(c7))/(k14+k16)
enddo                                   next arrangement
let c7=c2*c3                            now get the actual test statistic
```

```
let k30=sum(c7)/sum(c3)−(sum(c2)−sum(c7))/(k1−sum(c3))
let c9=c8−k30                          subtract from the arrangement
                                       test statistics
                                       and code zero or one
code (−1000000:−0.0000001)0 (0:1000000) 1 c9 c10
let c11=abs(c8)−abs(k30)               same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c11 c12
let c13(1)=k30                         display actual test statistic at the
                                       top of column 13
let c13(2)=sum(c10)                    count the arrangement statistics
                                       that are at least as large
let c13(3)=c13(2)/2001                 and find the one-tailed
                                       probability
let c13(4)=abs(k30)                    same for absolute values
let c13(5)=sum(c12)                    and find the two-tailed
let c13(6)=c13(5)/2001                 probability
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 9 for 2000 entry
Lines 2 and 5 up from bottom for 2001 entries

Location of Design 2 Test Results

After running, the test statistic (for the actual data) is at the top of Column 13 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 2 Example

Row	*Result*	*1st run c13*	*2nd run c13*	*3rd run c13*
1	One-tailed statistic	2.592	2.592	2.592
2	No. as large	84	76	97
3	One-tailed probability	0.042	0.038	0.049
4	Two-tailed statistic	2.592	2.592	2.592
5	No. as large	84	76	97
6	Two-tailed probability	0.042	0.038	0.049

Mean time for three runs=1 min, 25 sec

Statistical Conclusion for Design 2 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 2000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.042. Therefore, the obtained difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 3 (AB MULTIPLE BASELINE)

Specifications for Design 3 Example

Number of participant (or behavior) replications (i.e., baselines)	=	3
Total number of observation periods per participant (or behavior)	=	18
Minimum number of control periods before treatment starts	=	6
Minimum number of treatment periods	=	6

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention points for the example are 9, 12, and 8.

Commented Macro for Design 3 (Macro File Name: des3.txt)

```
gmacro                          collect information on number
des3                            of observations per participant,
let k1=c1(2)                    minimum number of
                                preintervention periods,
                                minimum number of intervention
                                periods, and number of
                                participants from column 1
let k2=c1(3)                    find the first and last possible
let k3=c1(4)                    intervention points
let k5=c1(1)
let k10=k2+1
let k11=k1−k3+1                 number of arrangements
do k20=1:2000                   find a random intervention
random k5 c5;                   point for each participant for this
                                arrangement
integer k10:k11.                empty column 8, which will be
erase c8                        reused for each arrangement
do k21=1:k5                     for each participant make a
                                column of zeros and ones for the
                                random intervention point
```

```
let k14=c5(k21)−1
let k15=k1−k14
set c6
k14(0) k15(1)
end
letc7=c6*(k14+kl5)/(k14*k15)−1/k14
```

turn the zeros and ones into the correct multipliers to get the difference between intervention and preintervention means

```
if k21=1
let c8=c7
else
stack c8 c7 c8
endif
```

stack the results for all participants for this arrangement

```
enddo
let c9=c2*c8
```

next participant
find the difference between means for this arrangement and sum over participants, then store

```
let c10(k20)=sum(c9)
enddo
do k21=1:k5
```

next arrangement
now to get the actual test statistic, first get a column with ones against the current participant and zeros elsewhere

```
let c6=c4−k21+1
code (−100:0)0 (2:100)0 c6 c6
let c7=c3*c6
let k15=sum(c7)
let c7=c7*k1/((k1−k15)*k15)−1/(k1−k15)
let c7=c7*c6
if k21=1
```

make a column that reproduces the zeros and ones for the current participant with zeros elsewhere
and count the intervention points for the current participant
turn the zeros and ones into the correct multipliers to get the difference between intervention and preintervention means
replace the zeros opposite the other participants
then put the correct multipliers for all participants into one column

```
let c8=c7
else
let c8=c8+c7
endif
```

```
enddo
let k30=sum(c2*c8)
let c11=c10−k30
```

next participant
find the actual test statistic
subtract from the arrangement test statistics and code zero or one

```
code (−1000000:−0.0000001)0 (0:1000000)1 c11 c12
let c13=abs(c11)−abs(k30)                    same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c13 c14
let c15(1)=k30                               display actual test statistic at the
                                             top of column 15
let c15(2)=sum(c12)                          count the arrangement statistics
let c15(3)=c15(2)/2001                       that are at least as large
                                             and find the one-tailed
                                             probability
let c15(4)=abs(k30)                          same for absolute values
let c15(5)=sum(c14)                          and find the two-tailed
let c15(6)=c15(5)/2001                       probability
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 9 for 2000 entry
Lines 2 and 5 up from bottom for 2001 entries

Location of Design 3 Test Results

After running, the test statistic (for the actual data) is at the top of Column 15 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large, and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 3 Example

Row	*Result*	*1st run c15*	*2nd run c15*	*3rd run c15*
1	One-tailed statistic	5.915	5.915	5.915
2	No. as large	74	67	72
3	One-tailed probability	0.037	0.034	0.036
4	Two-tailed statistic	5.915	5.915	5.915
5	No. as large	74	67	72
6	Two-tailed probability	0.037	0.034	0.036

Mean time for three runs=3 min, 1 sec

Statistical Conclusion for Design 3 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of three communication aid users would increase when a word prediction system was introduced, the proportion of 2000 randomly sampled data divisions giving a combined rate difference

in the predicted direction at least as large as the experimentally obtained difference was 0.037. Therefore, the obtained summed difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 4 (ABA MULTIPLE BASELINE)

Specifications for Design 4 Example

Number of participants	=	2
Total number of observation periods per participant	=	12
Minimum number of control periods before treatment starts	=	3
Minimum number of treatment periods	=	3
Minimum number of control periods after withdrawal of treatment	=	3

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pairs of intervention and withdrawal points for the example are as follows: Participant 1 at Periods 6 and 10; Participant 2 at Periods 4 and 8.

Commented Macro for Design 4 (Macro File Name: des4.txt)

```
gmacro                          collect information on number
des4                            of observations per participant,
let k1=c1(2)                    minimum number of
                                preintervention periods,
                                minimum number of intervention
                                periods, minimum number of
                                withdrawal periods, and number
                                of participants from column 1
let k2=c1(3)                    find the first and last possible
let k3=c1(4)                    intervention points
let k4=c1(5)
let k5=c1(1)
let k10=k2+1
let k11=k1−k3−k4+1              number of arrangements
do k20=1:2000                   empty column 9, which will be
erase c9                        reused for each arrangement
do k21=1:k5                     deal with each participant in each
random 1 c5;                    arrangement
integer k10:k11.                find a random intervention
                                point for each participant for this
                                arrangement
let k12=c5(1)+k3                find the first and last possible
                                withdrawal points
```

```
let k13=k1−k4+1
if k12=k13
let c6(1)=k1−k4+1
else
random 1 c6;                          find a random withdrawal point
                                      for this arrangement
integer k12:k13.                      for each participant make a column
endif                                 of zeros and ones for the
let k14=c5(1)−1                       random intervention and
let k15=c6(1)−c5(1)                   withdrawal points
let k16=k1−c6(1)+1
set c7
k14(0) k15(1) k16(0)
end
let c8=c7*(k14+k15+k16)/((k14+k16)*k15)−1/(k14+k16)
                                      turn the zeros and ones into the
                                      correct multipliers to get the
                                      difference between intervention
                                      and preintervention/withdrawal
                                      means
if k21=1
let c9=c8
else
stack c9 c8 c9                        stack the results for all
endif                                 participants for this
enddo                                 arrangement
                                      next participant
let c10=c2*c9                         find the difference between
                                      means for this arrangement and
                                      sum over participants, then store
let c11(k20)=sum(c10)                 next arrangement
enddo                                 now to get the actual test
do k21=1:k5                           statistic, first get a column with
                                      ones against the current
                                      participant and zeros elsewhere
let c7=c4−k21+1                       make a column that reproduces
code (−100:0)0 (2:100)0 c7 c7         the zeros and ones for the
let c8=c3*c7                          current participant with zeros
let k15=sum(c8)                       elsewhere
                                      and count the intervention points
```

```
let c8=c8*k1/((k1−k15)*k15)−1/(k1−k15)      for the current participant
let c8=c8*c7                                turn the zeros and ones into the
if k21=1                                    correct multipliers to get the
                                            difference between intervention
                                            and preintervention/withdrawal
                                            means
                                            replace the zeros opposite the
                                            other participants
                                            then put the correct multipliers
                                            for all participants into one
                                            column
let c9=c8
else
let c9=c9+c8
endif
enddo
let k30=sum(c2*c9)                          find the actual test statistic
let c12=c11−k30                             subtract from the arrangement
                                            test statistics and code zero or
                                            one
code (−1000000:−0.0000001)0 (0:1000000)1 c12 c13
let c14=abs(c11)−abs(k30)                   same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c14 c15
let c16(1)=k30                              display actual test statistic at the
                                            top of column 16
let c16(2)=sum(c13)                         count the arrangement statistics
let c16(3)=c16(2)/2001                      that are at least as large
                                            and find the one-tailed
                                            probability
let c16(4)=abs(k30)                         same for absolute values
let c16(5)=sum(c15)                         and find the two-tailed
let c16(6)=c16(5)/2001                      probability
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 10 for 2000 entry
Lines 2 and 5 up from bottom for 2001 entries

Location of Design 4 Test Results

After running, the test statistic (for the actual data) is at the top of Column 16 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 4 Example

Row	*Result*	*1st run c16*	*2nd run c16*	*3rd run c16*
1	One-tailed statistic	3.750	3.750	3.750
2	No. as large	48	38	55
3	One-tailed probability	0.024	0.019	0.028
4	Two-tailed statistic	3.750	3.750	3.750
5	No. as large	48	38	55
6	Two-tailed probability	0.024	0.019	0.028

Mean time for three runs=3 min, 34 sec

Statistical Conclusion for Design 4 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of two communication aid users would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 2000 randomly sampled data divisions giving a combined rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.024. Therefore, the obtained summed difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 5 (ONE-WAY SMALL GROUPS AND SINGLE-CASE RANDOMIZED TREATMENT)

Specifications for Design 5 Example

Total number of participants (or treatment occasions)	=	10
Number of treatments: communication aid systems (or levels of symbol translucency)	=	4
Number of participants per communication system (or symbol sets per translucency level)	=	2, 2, 3, 3

Commented Macro for Design 5 (Macro File Name: des5.txt)

```
gmacro
des5
let k1=c1(1)          collect the number of
brief=0               observations from column 1
do k20=1:2000         suppress the output from the
sample k1 c2 c4       one-way ANOVA command
                      number of arrangements
                      arrangements of the data into
                      column 4
```

```
oneway c4 c3 c5                    perform a one-way ANOVA and
                                   store the residuals in column 5
let c6(k20)=ssq(c5)                find the RSS for this
                                   arrangement and store it in
                                   column 6
enddo                              next arrangement
oneway c2 c3 c5                    now get the actual test statistic
let k30=ssq(c5)                    subtract the arrangement test
let c7=k30−c6                      statistics
                                   and code zero or one
code (−1000000:−0.0000001)0 (0:1000000)1 c7 c8
let c9(1)=k30                      display actual test statistic at the
let c9(2)=sum(c8)                  top of column 9
let c9(3)=c9(2)/2001               count the arrangement statistics
brief=2                            that are at least as small
endmacro                           and find the probability
                                   restore the default output code
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 5 for 2000 entry
Line 3 up from bottom for 2001 entry

Location of Design 5 Test Results

After running, the test statistic (for the actual data) is at the top of Column 9 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as small (because the test statistic is RSS) and below that is the two-tailed probability.

Randomization Test Results for Design 5 Example

Row	*Result*	*1st run c9*	*2nd run c9*	*3rd run c9*
1	Statistic	3.333	3.333	3.333
2	No. as small	11	4	8
3	Two-tailed probability	0.006	0.002	0.004

Mean time for three runs=29 sec

Statistical Conclusion for Design 5 (One-Way Small Groups) Example

In a randomization test of the prediction that four communication aids would differ in the time taken to learn to use them to a criterion, the proportion of 2000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.006. Therefore, the obtained differences in learning time with the four communication aids were statistically significant ($p<0.01$).

DESIGN 5a (SMALL GROUPS OR SINGLE-CASE—TWO RANDOMIZED TREATMENTS)

Specifications for Design 5a Example

Total number of participants (or treatment occasions)	=	9
Number of treatments: communication aid systems (or levels of symbol translucency)	=	2
Number of participants per communication system (or symbol sets per translucency level)	=	4, 5

For a one-tailed test the level with the higher expected mean is coded 2 and the level with the lower expected mean is coded 1.

Commented Macro for Design 5a (Macro File Name: des5a.txt)

```
gmacro                                  collect the number of
des5a                                   observations from column 1
let k1=c1(1)                            turn the group codes into zero
let c3=c3–1                             and one
do k20=1:2000                           number of arrangements
sample k1 c2 c4                         arrangements of the data into
let c5=c3*c4                            column 4
                                        find the difference between
                                        means for this arrangement, then
                                        store
let c6(k20)=sum(c5)/sum(c3)−(sum(c2)−sum(c5))/(k1−sum(c3))
enddo                                   next arrangement
let c5=c2*c3                            now get the actual test statistic
let k30=sum(c5)/sum(c3)−(sum(c2)−sum(c5))/(k1−sum(c3))
let c7=c6−k30                           subtract it from the
                                        arrangement test statistics and
                                        code zero or one
code (−1000000:−0.0000001)0 (0:1000000)1 c7 c8
let c9=abs(c6)−abs(k30)                 same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c9 c10
let c11(1)=k30                          display actual test statistic at the
                                        top of column 11
let c11(2)=sum(c8)                      count the arrangement statistics
let c11(3)=c11(2)/2001                  that are at least as large
                                        and find the one-tailed
                                        probability
```

```
let c11(4)=abs(k30)          same for absolute values
let c11(5)=sum(c10)          and find the two-tailed
let c11(6)=c11(5)/2001       probability
let c3=c3+1                  restore the group codes
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 5 for 2000 entry
Lines 3 and 6 up from bottom for 2001 entries

Location of Design 5a Test Results

After running, the test statistic (for the actual data) is at the top of Column 11 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 5a Example

Row	*Result*	*1st run c11*	*2nd run c11*	*3rd run c11*
1	One-tailed statistic	1.550	1.550	1.550
2	No. as large	86	87	65
3	One-tailed probability	0.043	0.043	0.032
4	Two-tailed statistic	1.550	1.550	1.550
5	No. as large	154	159	146
6	Two-tailed probability	0.007	0.079	0.073

Mean time for three runs=39 sec

Statistical Conclusion for Design 5a (One-Tailed Single-Case) Example

In a randomization test of the prediction that high-translucency symbols will take fewer sessions than low translucency symbols for a communication aid user to learn, the proportion of 2000 randomly sampled data divisions giving a learning sessions difference in the predicted direction at least as large as the experimentally obtained difference was 0.043. Therefore, the obtained difference in learning sessions required was statistically significant ($p<0.05$; one-tailed).

It may be noted that, had a directional prediction not been made in this instance, the two-tailed test would not have shown a significant difference ($p=0.077$).

DESIGN 6 (ONE-WAY SMALL GROUP REPEATED MEASURES AND SINGLE-CASE RANDOMIZED BLOCKS)

Specifications for Design 6 Example

Number of participants (or number of blocks)	=	3
Number of conditions	=	4
Total number of observations (conditions×participants or blocks)	=	12

Each participant must receive the same number of measures (or each treatment must appear once in each block).

Commented Macro for Design 6 (Macro File Name: des6.txt)

```
gmacro                          suppress the output from the
des6                            two-way ANOVA command
brief=0                         collect information on number
let k1=c1(1)                    of observations and number of
                                participants or blocks from
                                column 1
let k5=c1(2)                    find the number of treatments or
let k6=k1/k5                    measures
set c5                          make a list of the treatment
1:k6                            numbers to sample from
end
do k20=1:2000                   number of arrangements
erase c7                        empty column 7, which will be
do k21=1:k5                     reused for each arrangement
sample k6 c5 c6                 deal with each participant or
                                block in each arrangement
                                arrangements of the treatment
                                numbers within each block and
                                collect in column 7
if k21=1
let c7=c6
else
stack c7 c6 c7                  next participant or block
endif                           do a two-way ANOVA and put
enddo                           the residuals in column 8
twoway c2 c4 c7 c8              find the RSS for this arrangement
let c9(k20)=ssq(c8)             and store
enddo                           next arrangement
twoway c2 c4 c3 c8              now to get the actual test statistic
let k30=ssq(C8)                 subtract the arrangement test
let c10=k30−c9                  statistics from it and code zero
                                or one
```

```
code (−1000000:−0.0000001)0 (0:1000000)1 c10 c11
let c12(1)=k30                    display actual test statistic at the
let c12(2)=sum(c11)               top of column 12
let c12(3)=c12(2)/2001            count the arrangement statistics
brief=2                           that are at least as small
endmacro                          and find the probability
                                  restore the default output code
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 10 for 2000 entry
Line 3 up from bottom for 2001 entry

Location of Design 6 Test Results

After running, the test statistic (for the actual data) is at the top of Column 12 in the worksheet. In the cell belows is the count of arrangements for which the statistic is at least as small (because the test statistic is RSS) and below that is the two-tailed probability.

Randomization Test Results for Design 6 Example

Row	*Result*	*1st run c12*	*2nd run c12*	*3rd run c12*
1	Statistic	98.500	98.500	98.500
2	No. as small	4	3	4
3	Two-tailed probability	0.002	0.002	0.002

Mean time for three runs=1 min, 32 sec

Statistical Conclusion for Design 6 (One-Way Small Groups) Example

In a randomization test of the prediction that the number of initiations made by communication aid users will differ depending on the level of experience of conversational partners, the proportion of 2000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.002. Therefore, the obtained differences in number of user initiations with partners with different levels of experience was statistically significant ($p<0.01$).

DESIGN 6a (TWO REPEATED MEASURES ON SMALL GROUP OR SINGLE-CASE BLOCKS)

Specifications for Design 6a Example

Number of participants (or number of blocks)	=	7
Number of conditions	=	2

Each participant must receive the same number of measures (or each treatment must appear once in each block).

For a one-tailed test the condition with the higher expected mean is coded 2 and the condition with the lower expected mean is coded 1.

Commented Macro for Design 6a (Macro File Name: des6a.txt)

```
gmacro
des6a
let k5=c1(1)                    collect information on number of
do k20=1:2000                   participants or blocks from
letk1=2*k5                      column 1
random k1 c5;                   number of arrangements
                                find the number of observations
                                make a column of random
                                numbers between zero and one
                                (only half will be used)
uniform 0 1.                    for each participant or block we
do k21=1:k5                     need a random arrangement of
let k23=2*k21−1                 the condition codes
let k24=k23+1                   get the row numbers for the
let c6(k23)=round(c5(k23), 0)   current participant or block
let c6(k24)=1−c6(k23)           turn the first random number
enddo                           for this participant or block into
let c7(k20)=(2*sum(c6*c2)−sum(c2))/k5   0 or 1
enddo                           for this participant or block, the
let k30=(2*sum((c3−1)*c2)−sum(c2))/k5   second condition code is
let c8=c7−k30                   whichever the first is not
                                next participant
                                find the test statistic for this
                                arrangement and store it
                                next arrangement
                                find the actual test statistic
                                subtract from the arrangement
                                test statistics and code zero or
                                one
code (−1000000:−0.0000001)0 (0:1000000)1 c8 c9
let c10=abs(c7)−abs(k30)        same for absolute values
code (−1000000:−0.0000001)0 (0:1000000)1 c10 c11
et c12(1)=k30                   display actual test statistic at the
let c12(2)=sum(c9)              top of column 12
let c12(3)=c12(2)/2001          count the arrangement statistics
let c12(4)=abs(k30)             that are at least as large
let c12(5)=sum(c11)             and find the one-tailed
let c 2(6)=c12(5)/2001          probability
endmacro                        same for absolute values
                                and find the two-tailed
                                probability
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 4 for 2000 entry
Lines 2 and 5 up from bottom for 2001 entries

Location of Design 6a Test Results

After running, the test statistic (for the actual data) is at the top of Column 12 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large, and below that is the one-tailed probability. The absolute value of the test statistic, the count of arrangement statistics that are at least as large in absolute value, and the two-tailed probability follow below.

Randomization Test Results for Design 6a Example

Row	*Result*	*1st run c12*	*2nd run c12*	*3rd run c12*
1	One-tailed statistic	1.857	1.857	1.857
2	No. as large	84	88	95
3	One-tailed probability	0.042	0.044	0.047
4	Two-tailed statistic	1.857	1.857	1.857
5	No. as large	177	189	185
6	Two-tailed probability	0.088	0.094	0.092

Mean time for three runs=4 min, 3 sec

Statistical Conclusion for Design 6a (One-Tailed Single-Case) Example

In a randomization test of the prediction that a communication aid user will choose to use their high-tech aid more frequently with a speech and language therapist than with a family member, the proportion of 2000 randomly sampled data divisions giving a difference in high-tech aid use in the predicted direction at least as large as the experimentally obtained difference was 0.042. Therefore, the obtained difference in high-tech use was statistically significant *(p*<0.05, one-tailed).

As in Design 5a, if a directional prediction had not been made in this instance, the two-tailed test would not have shown a significant difference *(p*=0.088).

DESIGN 7 (TWO-WAY FACTORIAL SINGLE CASE)

Specifications for Design 7 Example

Factor 1=interface device (touch-screen coded 1, joystick coded 2)
Factor 2=display mode (static coded 1, dynamic coded 2)

Total number of observations	=	16
Number of observations per condition (must be equal)	=	4

Directional prediction for Factor 1: joystick>touch-screen
Directional prediction for Factor 2: dynamic>static
Predicted interaction: Rate slowest for touch-screen interface with dy¬ namic display (see Fig 5.1)
Predictions for simple effects based on predicted interaction:
dynamic>static with touch-screen interface
joystick>touch-screen with static display
dynamic will not differ significantly from static with joystick interface
joystick will not differ significantly from touch-screen with dynamic display

Commented Macro for Design 7 (Macro File Name: des7.txt)

gmacro des7	
brief=0	**suppress the output from the two-way ANOVA command**
let k1=c1(1)	**collect information on number of observations from column 1**
let k7=k1/2	**find the number of observations at each level of a factor**
let k8=k1/4	**and the number of observations at each combination of factor levels**
set c5	**make a list of zeros and ones to sample from**
(0:1)k8 end	
sort c2−c4 c2−c4;	**to work on the main effect of factor 1, sort the data into blocks by factor 2 levels**
by c4. do k20=1:2000 erase c7	**number of arrangements for factor 1 empty column 7, which will be reused for each arrangement**
do k22=1:2 sample k7 c5 c6	**deal with each level of factor 2 arrange the factor 1 levels within each level of factor 2 (but subtract one from level to ease calculation of means); collect in column 7**
if k22=1 let c7=c6 else stack c7 c6 c7 endif	
enddo let c8(k20)=(2*sum(c7*c2)−sum(c2))/k7	**next level of factor 2 find the test statistic for factor 1 and this arrangement and store it**
enddo Ietk30=(2*sum(c2*(c3−1))−sum(c2))/k7 sort c2−c4 c2−c4;	**next arrangement for factor 1 find the actual test statistic for factor 1 to work on the main effect of factor 2, sort the data into blocks by factor 1 levels**
by c3. do k20=1:2000 erase c7	**number of arrangements for factor 2 empty column 7, which will be reused for each arrangement**

```
do k22=1:2 sample k7 c5 c6                    deal with each level of factor 1 arrange
                                              the factor 2 levels within each level of
                                              factor 1 (but subtract one from level to
                                              ease calculation of means); collect in
                                              column 7
if k22=1 let c7=c6 else
stack c7 c6 c7 endif
enddo let C9(k20)=(2*sum(c7*c2)−sum(c2))/k7   next level of factor 1 find the test statistic
                                              for factor 2 and this arrangement and
                                              store it
enddo let k31=(2*sum(c2*(c4–1))−sum(c2))/k7   next arrangement for factor 2 find the
                                              actual test statistic for factor 1
let c10=c8-k30                                subtract actual from the arrangement
                                              test statistics and code zero or one
let c11=c9-k31
code (−1000000:−0.00001)0 (0:1000000)1 c10 c12
code (−1000000:−0.00001)0 (0:1000000)1 c11 c13
let c14=abs(c8)−abs(k30) let                  same for absolute values
c15=abs(c9)−abs(k31)
code (−1000000:−0.00001)0 (0:1000000)1 c14 c16
code (−1000000:−0.00001)0 (0:1000000)1 c15 c17
let c18(1)=k30                                display actual factor 1 test statistic at the
                                              top of column 18
let c18(2)=sum(c12)                           count the arrangement statistics that are
                                              at least as large
let c18(3)=c18(2)/2001                        and find the one-tailed probability
let c18(4)=abs(k30) let c18(5)=sum(c16)       same for absolute values
let c18(6)=c18(5)/2001                        and find the two-tailed probability
let c18(7)=k31 let c18(8)=sum(c13) let c      same for factor 2
8(9)=c1 8(8)/2001 let c18(10)=abs(k31) let
c18(11)=sum(c17) let c18(12)=c18(11)/2001
brief=2 endmacro                              restore the default output code
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Lines 12 and 27 for 2000 entries
Lines 3, 6, 9, and 12 up from bottom for 2001 entries

Location of Design 7 Test Results

After running, Column 18 contains the results for tests of the main effects. At the top is the one-tailed test statistic for Factor 1 (the mean of Level 2–the mean of Level 1). Next is the number of arrangement statistics that are at least as large, then the one-tailed probability. Next is the absolute value of the test statistic, followed by the number of arrangement statistics that are at least as large in absolute value, and then the two-tailed probability. Rows 7 to 9 in Column 18 have the one-tailed results for Factor 2, then Rows 10 to 12 contain the two-tailed results for Factor 2. There is no randomization test for the interaction (see Design 7 in chap. 5). To examine simple effects, use Design 5a. An example for one of the simple effects is provided later.

Randomization Test Results for Design 7 Example

Row	*Result*	*1st run c18*	*2nd run c18*	*3rd run c18*
	Main Effect for Factor 1 (Interface Device)			
1	One-tailed statistic	1.000	1.000	1.000
2	No. as large	248	215	206
3	One-tailed probability	0.124	0.107	0.103
4	Two-tailed statistic	1.000	1.000	1.000
5	No. as large	458	414	408
6	Two-tailed probability	0.229	0.207	0.204
	Main Effect for Factor 2 (Display Mode)			
7	One-tailed statistic	1.500	1.500	1.500
8	No. as large	61	60	71
9	One-tailed probability	0.030	0.030	0.035
10	Two-tailed statistic	1.500	1.500	1.500
11	No. as large	117	125	117
12	Two-tailed probability	0.058	0.062	0.058

Mean time for three runs=2 min, 23 sec

Statistical Conclusions for Design 7 (One-Tailed Main Effects) Example

Randomization tests of the main effects in a 2×2 factorial experiment on a single communication aid user were carried out. In a test of the prediction that rate of communication would be faster when the interface device was a joystick rather than a touch-screen, the proportion of 2000 randomly sampled data divisions giving a difference in the predicted

direction at least as large as the experimentally obtained difference was 0.124. Therefore, the main effect of interface device was not statistically significant ($p>0.05$; one-tailed). In a test of the prediction that rate of communication would be faster when a dynamic rather than a static display mode was used, the proportion of 2000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.030. Therefore, the main effect of display mode was statistically significant ($p<0.05$; one-tailed).

Testing Simple Effects in the Factorial Design

Although there is no randomization test available for testing the interaction between interface device and display mode, if a particular form of interaction has been predicted, predictions for simple effects can be derived from the predicted interaction, and these can be tested using a randomization test. These are, of course, precisely the follow-up tests that would normally be made following the finding of a significant interaction. The tests of simple effects can be carried out using the macro for Design 5a. We present the results for a randomization test of one of the four simple effects, that predicting that a dynamic display mode will be superior to a static display mode when a touch-screen interface is used.

The data for entry into the Design 5a worksheet would be as follows. At the top of Column 1, the number of observations involving only the touch-screen interface (i.e., 8) is entered. Column 2 will contain the touch-screen data for static and dynamic display modes and the display mode codes will be entered in Column 3, where the display mode predicted to result in a faster communication rate (i.e., the dynamic mode) is coded 2. The worksheet entries are shown in the Design 5a Worksheet Box (Simple Effects Example) under "Example for Design 7" in chapter 5. The file name of the worksheet on the CD-ROM is des5a_simple.mtw.

Randomization Test Results for Design 7 (Simple Effect of Display Mode With Touch-Screen Interface) Example

Row	*Result*	*1st run c11*	*2nd run c11*	*3rd run c11*
1	One-tailed statistic	3.250	3.250	3.250
2	No. as large	31	31	28
3	One-tailed probability	0.016	0.016	0.014
4	Two-tailed statistic	3.250	3.250	3.250
5	No. as large	56	56	58
6	Two-tailed probability	0.028	0.028	0.029

Mean time for three runs=38 sec

Statistical Conclusion for One-Tailed Test of a Simple Effect

Predictions for simple effects were derived from the predicted form of the interaction between interface device and display mode. In a randomization test of the prediction that

a dynamic display mode would result in a faster communication rate than a static display mode when the interface device was a touch-screen, the proportion of 2000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.016. Therefore, the simple effect of dynamic display with a touch-screen interface was statistically significant ($p<0.05$; one-tailed).

DESIGN 8 (ORDINAL PREDICTIONS WITHIN NONEXPERIMENTAL DESIGNS)

The randomization test that we present for application to a range of designs involving ordinal predictions is only valid in a strict sense when a genuine random assignment procedure has been incorporated in the design. We believe that there may be some circumstances in which use of the randomization test is justified even though a genuinely random procedure for the assignment of experimental units to observation periods has not been followed. We trust that those interested in using the test will read our discussion of the issue in chapter 10 and come to a view of the appropriateness of the test for analysis of their own data.

Example data for a predicted unique order and for a predicted partial order were presented in chapter 5—each prediction is tested in a sepa-rate run of the macro. The file names of the worksheets on the CD-ROM for the unique and partial predictions are *des8unique.mtw* and *des8partial.mtw,* respectively.

Specifications for Design 8 Example

Number of participants (or number of observations on a single participant) = 6

Predicted order: Unique or Partial

Commented Macro for Design 8 (Macro File Name: des8.txt)

```
gmacro des8 let k1=c1(1)        collect information on number of observations from column 1
do k20=1:2000 sample            number of arrangements arrangements of the data into column 4
k1 c2 c4
Ietc5(k20)=sum(c3*c4)           find the test statistic for the arrangement and store next arrange-
enddo                           ment now get the actual test statistic subtract from the arrange-
let k30=sum(c2*c3)              ment test statistics and code zero or one
let c6=c5-k30
code (-1000000:-0.0000001)0 (0:1000000)1 c6c7
let c8(1)=k30                   display actual test statistic at the top of column 8 count the
let c8(2)=sum(c7)               arrangement statistics that are at least as large and find the prob-
let c8(3)=c8(2)/2001            ability
endmacro
```

Location of Sample Size (2000 and 2001 Entries) in Macro

Line 4 for 2000 entry
Line 2 up from bottom for 2001 entry

Location of Design 8 Test Results

After running, the test statistic (for the actual data) is at the top of Column 8 in the worksheet. In the cell below is the count of arrangements for which the statistic is at least as large, and below that is the one-tailed probability.

Randomization Test Results for Design 8 Example

Row	*Result*	*1st run c8*	*2nd run c8*	*3rd run c8*
	Test of the Unique Order Prediction			
1	One-tailed statistic	110.000	110.000	110.000
2	No. as large	28	41	26
3	One-tailed probability	0.014	0.020	0.013
	Test of the Partial Order Prediction			
1	One-tailed statistic	44.000	44.000	44.000
2	No. as large	218	211	177
3	One-tailed probability	0.109	0.105	0.088

Mean time for six runs=22 sec

Statistical Conclusion for Design 8 (Unique Order Prediction of Small Group Data) Example

In a randomization test of the prediction of a unique ordering of communicative competence ratings of communication aid users, the proportion of 2000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.014. Therefore, the correlation between the obtained order and the predicted order (based on the etiology of speech impairment and severity of additional physical impairments) was statistically significant ($p<0.05$; one-tailed).

Statistical Conclusion for Design 8 (Partial Order Prediction of Single-Case Data) Example

In a randomization test of the prediction of a partial ordering of frequency of questions asked by a communication aid user of same-sex and opposite-sex partners, the proportion of 2000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.109. Therefore, the correlation between the obtained order and the predicted order (based on the gender of partners) was not statistically significant ($p>0.05$, one-tailed).

Chapter 7
Randomization Tests Using Excel

All of the designs described in chapter 5 can be subjected to randomization tests within Microsoft Excel using the same general procedures. We suggest that you begin by copying all of the files in the Excel subdirectory on the companion CD-ROM to a convenient directory. The files are Excel workbooks with .xls extensions (*design1.xls* to *design8unique.xls*). Each workbook comprises a macro that contains the commands necessary to run a randomization test and a data worksheet with the example data from chapter 5 entered. To run a macro, the workbook first has to be opened in Excel. It will be necessary to respond to the warning about macro viruses by clicking on the Enable Macros button. Opening the workbook will result in the example data being displayed in the worksheet (Sheet 1). The next step is to edit the worksheet so that it contains your own data, remaining in conformity with the worksheet specifications given in chapter 5. To edit a worksheet, simply replace the numerical entries in all columns with your own values, making sure that you leave the labels in the first row unchanged. Some of the macros require an initially empty column and some cells of the worksheet to be named, but column names have already been inserted where they are needed (labeled *both* in Designs 3 and 4, and *arrange* in Designs 5 and 5a) and the cell numbers are automatically adjusted for your own data by Excel, so you do not need to worry about these extra names. When you have entered your own data in the worksheet, you may wish to use *File>Save As* to store it under a new name. Then, when you close the workbook, you can say you don't wish to save the changes you made, and you will still have the original example data and macro under the original name.

To run the macro once your data are in the worksheet, select from the menu bar *Tools>Macro>Macros* and from the dialog box select *Run* (there is only one macro in the workbook so you don't need to select which one to run). All of the macros have been set to do 1000 data arrangements. With no other packages running concurrently, most of them take less than 5 min with the example data on a Pentium II or III computer, but there are a few exceptions, including two designs (Designs 3 and 4) that take substantially longer. You can run more or fewer arrangements if you edit two or three lines of the macro. To do this, choose *Edit* instead of *Run* while in the macros dialog box, and you will see the Visual Basic code. Look for the line saying *For j=1 To 1000,* and change the 1000 to the number you want (say, 2000). The probability will now be incorrect unless you also change the (1000+1) in the denominator of the line(s) that calculates the probability to the same new value, for example, (2000+1). If there is both a one- and a two-tailed probability this will have to be done for both. The lines containing (1000+1) are near the end of the macro. The annotated macro listings will be helpful in identifying these lines if you want to change them, or you can use *Edit>Replace* from the top menu bar to locate and change the 1000 and 1000+1 entries. To return to the worksheet after editing the macro, select *File>Close and Return to Microsoft Excel*. You may find it useful, as we did, to check the running of the macro by changing to 10 arrangements, without bothering to correct the probability. This gives a very quick check, and if all goes well, change back to 1000 and do the real run.

In the remainder of the chapter, the following are listed for each design:

1. The design specifications for the example in chapter 5.
2. The macro for the design, with bolded comments indicating the function of each section of the macro.
3. The location of the randomization test results on the worksheet when the macro has been run.
4. The randomization test results for three runs of the macro on the example worksheet, including the average time taken for the run using Excel on a Pentium P266 MHZ with 64 Mb RAM and an Intel processor, with no other packages running concurrently.
5. A statement of the statistical conclusion based on the first run of the macro (which normally would be the only result obtained).

It should be noted that the count of rearrangement statistics that are at least as large (or in some cases, at least as small) as the test statistic, and its probability under the null hypothesis, will vary slightly from run to run on the same data. This is due to random sampling fluctuations as 1000 arrangements are randomly sampled. Provision of the results of three runs of the macro should help users to gain some idea of the extent of variation using 1000 samples. We have generally provided sample data that would give probabilities close to a critical value (e.g., p =0.05) if systematic sampling of all possible arrangements were generated. For example, in Design 1, systematic generation of all 21 possible arrangements of the data would yield a probability of 1/21=0.048. Consequently, using random sampling of arrangements with replacement, the statistical decisions obtained sometimes differ between runs with the same data. If your own data produce probabilities close to a critical value, you may wish to obtain more stable probability estimates than those provided by 1000 samples. However, the propriety of sequential testing of this kind in the absence of taking any steps to protect the significance level is open to question and, ideally, a decision about how many samples to use should be made at the outset (see the discussion of this issue in chap. 12). If you nonetheless decide to pursue a sequential strategy or you decide at the outset to use a different number of samples, you should open the macro using Edit as described earlier, locate the 1000 and 1001 entries and change them to, say, 5000 and 5001. You can, of course, save the edited macro with a new name if you wish. Increasing the number of random samples will of course increase the length of the run time, so it may be a good idea to try a run, possibly using simulated data comparable to your real data, with the original 1000 rearrangements on your own computer to see how long that takes before making a change.

It should also be noted that in some of the designs, one- and two-tailed probabilities often, but not necessarily, will be identical. For example in the single-case AB design (Design 1), the probabilities will only differ when one or more arrangements of the data produces a difference between means in the nonpredicted direction that is at least as large as the observed difference between means.

DESIGN 1 (AB)

Specifications for Design 1 Example

Total number of observation periods	=	36
Minimum number of baseline periods	=	8
Minimum number of treatment periods	=	8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention point for the example is at Period 15.

Commented Macro for Design 1 (Macro File Name: design1.xls)

```
Sub design1()
Columns("D:O").ClearContents                    clear area to be used
Range("A5").Select                              calculate last possible
                                                intervention point
ActiveCell.FormulaR1C1="=R2C1−R4C1+1"
Range("A6").Select                              calculate number of available
                                                intervention points
ActiveCell.FormulaR1C1="=R5C1−R3C1"
Dim i As Integer
Dim last_i As Integer                           number of observations
last_i=Range("A2")
Dim lastrow$                                    row number for bottom of
                                                columns of data and random
                                                intervention codes
lastrow$=last_i+1
Dim lastperm$                                   row number for bottom of
lastperm$=Range("A6")+1                         columns of random
                                                intervention points
Range("D2").Select                              list the observation numbers
ActiveCell.FormulaR1C1 ="1"                     from 1 to "A2"
Range("D3").Select
ActiveCell.FormulaR1C1 ="2"
Range("D2: D3").Select
Selection.AutoFill Destination:=Range("D2:D" & lastrow$), _
Type:=xlFillDefault                             fill in a list of cumulative
Range("E2").Select                              probabilities for the permitted
                                                intervention points
ActiveCell.FormulaR1C1="=(RC[−1]−1)/R6C1"
Selection.AutoFill Destination:=Range("E2:E" & lastperm$), _
Type:=xlFillDefault                             list the possible intervention
Range("F2").Select                              points
```

```
ActiveCell.FormulaR1C1="=RC[−2]+R3C1"
Selection.AutoFill Destination:=Range("F2:F" & lastperm$), _
Type:=xlFillDefault
Columns("E:F").Select                          make a lookup table with the
                                               probabilities and possible
                                               intervention points
ActiveWorkbook.Names.Add Name:="lookup", RefersToR1C1:="=C5:C6"
Dim j As Integer                               number of randomly generated
For j=1 To 1000                                intervention points
Range("G2"). Select                            find a random number and fill
ActiveCell.FormulaR1C1="=RAND()"               the column with it
Selection.Copy
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
Selection.AutoFill Destination:=Range("G2:G" & lastperm$), _
Type:=xlFillDefault                            look up the intervention point
Range("H2").Select                             for this random number
ActiveCell.FormulaR1C1="=VLOOKUP(RC[−1],lookup,2)"
Range("l2").Select                             make a column of zeros and
                                               ones corresponding to this
                                               intervention point
ActiveCell.FormulaR1C1="=IF(R2C8>RC[−5],0,1)"
Selection.AutoFill Destination:=Range("l2:l" & lastrow$),_
Type:=xlFillDefault                            sum the observations from
Range("J2").Select                             intervention for this
                                               intervention point
ActiveCell.FormulaR1C1=_
"=SUMIF(C[−1]:C[−1],""">0""",C[−8]:C[−8])"      and sum the observations
Range("J3").Select                             before intervention
ActiveCell.FormulaR1C1=_
"=SUMIF(C[−1]:C[−1],""1=0""",C[−8]:C[−8])"      count the observations before
Range("J4").Select                             intervention
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[−1]:C[−1],""">0"")"                 and count the observations from
Range("J5").Select                             intervention
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[−1]:C[−1],""="0"")"                 calculate the test statistic for this
Range("K2").Select                             intervention point
ActiveCell.FormulaR1C1 C1="=RC[−1]/R[2]C[−1]−R[1]C[−1]/R[3]C[−1]"
```

```
Selection.Copy                                          and store it
Range("L2").Select
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                                    generate the next random
                                                        intervention point
lastj$=j+1                                              last row of arrangement
                                                        statistics
Range("M2").Select                                      absolute value of arrangement
                                                        statistics
ActiveCell.FormulaR1C1="=ABS(C[-1])"
Selection.AutoFill Destination:=Range("M2:M" & lastj$), _
Type:=xlFillDefault
Range("N2").Select                                      now deal with the actual
                                                        experiment: sum the
                                                        observations from intervention
ActiveCell.FormulaR1C1=_
"=SUMIF(C[-11]:C[-11],""">O""",C[-12]:C[-12])"
Range("N4").Select                                      sum the observations before
ActiveCell.FormulaR1C1=_                                intervention
"=SUMIF(C[-11]:C[-11],""=0"",C[-12]:C[-12])"
Range("N4").Select                                      count the observations from
ActiveCell.FormulaR1C1=__                               intervention
"=COUNTIF(C[-11]:C[-11],"">0"")"
Range("N5").Select                                      count the observations before
                                                        intervention
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[-11]:C[-11],""=0"")"
Range("02").Select                                      calculate the test statistic for the
                                                        actual experiment
ActiveCell.FormulaR1C1="=RC[-1]/R[2]C[-1]-R[1]C[-1]/R[3]C[-1]"
Range("O3").Select                                      count the randomly generated
                                                        ones that are at least as big as
                                                        the actual one
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[-3]:C[-3],"">=""&R[-1]C)"                   and calculate the one-tailed
Range("O4").Select                                      probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("O5").Select                                      absolute value of actual test
                                                        statistic
ActiveCell.FormulaR1C1="=ABS(R[-3]C)"
```

```
Range("O6").Select                    count the randomly generated
                                      ones that are at least as big in
                                      absolute value as the absolute
                                      value of the actual one
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[−2]:C[−2]J""">=""&R[−1]C)"
Range("O7").Select                    and calculate the
                                      two-tailed probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
End Sub
```

Location of Design 1 Test Results

After running, the test statistic (for the actual data) is in Cell O2. In O3 is the count of arrangement test statistics at least as large and in O4 is the one-tailed probability. Cell O5 contains the absolute value of the test statistic, O6 contains the count of arrangement statistics that are at least as large in absolute value, and O7 contains the two-tailed probability.

Randomization Test Results for Design 1 Example

Row	*Result*	*1st run Col.O*	*2nd run Col.O*	*3rd run Col.O*
2	One-tailed statistic	2.656	2.656	2.656
3	No. as large	36	44	42
4	One-tailed probability	0.037	0.045	0.043
5	Two-tailed statistic	2.656	2.656	2.656
6	No. as large	36	44	42
7	Two-tailed probability	0.037	0.045	0.043

Mean time for three runs=4 min, 2 sec

Statistical Conclusion for Design 1 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would increase when a word prediction system was introduced, the proportion of 1000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.037. Therefore, the obtained difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 2 (ABA REVERSAL)

Specifications for Design 2 Example

Total number of observation periods	=	36
Minimum number of control periods before treatment starts	=	8

Minimum number of treatment periods	=	8
Minimum number of control periods after withdrawal of treatment	=	8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pair of intervention and withdrawal points for the example is at Periods 12 and 25.

Commented Macro for Design 2 (Macro File Name: design2.xls)

```
Sub design2()                                          clear area to be used
Columns("D:S").ClearContents                           calculate number of possible
Range("A6").Select                                     intervention points
ActiveCell.FormulaR1C1="=(R2C1−R3C1−R4C1−R5C1+1)"
Range("A7").Select                                     calculate number of possible
                                                       intervention and withdrawal
                                                       pairs
ActiveCell.FormulaR1C1="=R6C1*(R6C1+1)/2"
Dim last_i As Integer
last_i=Range("A2")                                     number of observations
Dim lastrow$                                           row number for bottom of data
lastrow$=last_i+1                                      columns and random
                                                       intervention and withdrawal
                                                       codes
Dim lastperm$                                          row number for bottom of
lastperm$=Range("A7")+1                                columns of random
                                                       intervention and withdrawal
                                                       pairs
Range("D2").Select                                     list the numbers of the possible
                                                       intervention and withdrawal
                                                       pairs
ActiveCell.FormulaR1C1="1"
Range("D3"). Select
ActiveCell.FormulaR1C1="2"
Range("D2:D3").Select
Selection.AutoFill Destination:=Range("D2:D" & lastperm$),_
Type:=xlFillDefault
Range("E2").Select                                     columns E, F, and G enable us
                                                       to list the possible
                                                       intervention and withdrawal
                                                       pairs in columns I and J
ActiveCell.FormulaR1C1="=INT(0.5+SQRT(2*RC[−1]))"
Selection.AutoFill Destination:=Range("E2:E" & lastperm$), _
Type:=xlFillDefault
Range("F2").Select
```

```
ActiveCell.FormulaR1C1="=INT((SQRT(8*(RC[−2]−1)+1)−1)/2)"
Selection.AutoFill Destination:=Range("F2:F" & lastperm$), _
Type:=xlFillDefault
Range("G2").Select
ActiveCell.FormulaR1C1="=RC[−3]−RC[−1]*(RC[−1]+1)/2"
Selection.AutoFill Destination:=Range("G2:G" & lastperm$), _
Type:=xlFillDefault
Range("l2").Select
ActiveCell.FormulaR1C1="=R2C1−R4C1−R5C1+1−RC[−3]"
Selection.AutoFill Destination:=Range("l2:l" & lastperm$), _
Type:=xlFillDefault
Range("J2").Select
ActiveCell.FormulaR1C1="=R2C1−R5C1+2−RC[−3]"
Selection.AutoFill Destination:=Range("J2:J" & lastperm$), _
Type:=xlFillDefault                                    column H has the cumulative
Range("H2").Select                                     probabilities for the possible
                                                       pairs
ActiveCell.FormulaR1C1="=(RC[−4]−1)/R7C1"
Selection.AutoFill Destination:=Range("H2:H" & lastperm$), _
Type:=xlFillDefault                                    make a lookup table with
Columns("H:J").Select                                  probabilities and
                                                       intervention and withdrawal
                                                       pairs
ActiveWorkbook.Names.Add Name:="lookup", RefersToR1C1:="=C8:C10"
Dim j As Integer                                       number of randomly
For j=1 To 1000                                        generated
                                                       intervention and withdrawal
                                                       pairs
Range("K2"). Select
ActiveCell.FormulaR1C1="=RAND()"                       find a random number and fill
                                                       a column with it
Selection.Copy
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _
SkipBlanks:=False, Transpose:=False
Selection.AutoFill Destination:=Range("K2:K" & lastperm$)
Range("L2").Select                                     look up the intervention and
                                                       withdrawal points
ActiveCell.FormulaR1C1="=VLOOKUP(RC[−1],lookup,2)"
Range("L3").Select
ActiveCell.FormulaR1C1="=VLOOKUP(RC[−1],lookup,3)"
Range("M2").Select                                     make a column of ones and
                                                       zeros corresponding to this pair
```

```
ActiveCell.FormulaR1C1=_
"=IF(RC[−9]<R2C12,0,IF(RC[−9]>=R3C12, 0, 1))"
Selection.AutoFill Destination:=Range("M2:M" & lastrow$), _
Type:=xlFillDefault
Range("N2").Select
ActiveCell.FormulaR1C1=_                          sum the observations from
"=SUMIF(C[−1]:C[−1],""">0""",C[−12]:C[−12])"       intervention to withdrawal
Range("N3").Select                                sum the observations before
ActiveCell.FormulaR1C1=_
"=SUMIF(C[−1]:C[−1]>""""=0""",C[−12]:C[−12])"
                                                  intervention and after
                                                  withdrawal
Range("N4").Select
ActiveCell.FormulaR1C1=_                          count the observations from
"=COUNTIF(C[−1]:C[−1],""">0""")"                  intervention to withdrawal
Range("N5"). Select
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[−1]:C[−1],""""=0""")"                 count the observations before
                                                  intervention and after
                                                  withdrawal
Range("O2").Select                                calculate the test statistic for
                                                  this pair
ActiveCell.FormulaR1C1="=RC[−1]/R[2]C[−1]−R[1]C[−1]/R[3]C[−1]"
Selection.Copy
Range("P2").Select                                and store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues            generate next random
Next                                              intervention and withdrawal pair
lastj$=j+1                                        last row of arrangement
Range("Q2").Select                                statistics
                                                  absolute value of arrangement
                                                  statistics
ActiveCell.FormulaR1C1="=ABS(C[−1])"
Selection.AutoFill Destination:=Range("Q2:Q" & lastj$), _
Type:=xlFillDefault                               now deal with the actual
Range("R2").Select                                experiment: sum the
                                                  observations from intervention
                                                  to withdrawal
ActiveCell.FormulaR1C1=_
"=SUMIF(C[−15]:C[−15],""">0""",C[−16]:C[−163)"     sum the observations before
Range("R3").Select                                intervention and after
                                                  withdrawal
```

```
ActiveCell.FormulaR1C1=_
"=SUMIF(C[−15]: C[−15],""=0"",[−16]:c[−16])"
Range("R4").Select
ActiveCell.FormulaR1C1=
```
count the observations from intervention to withdrawal

```
"=COUNTIF(C[−15]:C[−15],"">0"")"
Range("R5").Select
```
count the observations before intervention and after withdrawal

```
ActiveCell.FormulaR1C1=_
"=COUNTIF(C[−15]:C[−15],""=0"")"
Range("S2").Select
```
calculate the test statistic for the actual experiment

```
ActiveCell.FormulaR1C1="=RC[−1]/R[2]C[−1]−R[1]C[−1]/R[3]C[−1]"
Range("S3").Select
```
count the arrangement statistics that are at least as big as the actual one

```
ActiveCell.FormulaR1C1="=COUNTIF(C[−3]:C[−3],"">="" &R[−1]C)"
Range("S4").Select
```
and calculate the one-tailed probability

```
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
Range("S5").Select
```
calculate the absolute value of the test statistic for the actual experiment

```
ActiveCell.FormulaR1C1="=ABS(R[−3]C)"
Range("S6").Select
```
count the arrangement statistics that are at least as big in absolute value as the absolute value of actual one

```
ActiveCell.FormulaR1C1="=COUNTIF(C[−2]:C[−2]J"">=""&R[−1]C)"
Range("S7").Select
```
and calculate the two-tailed probability

```
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
End Sub
```

Location of Design 2 Test Results

After running, the test statistic (for the actual data) is in Cell S2. In S3 is the count of arrangement test statistics at least as large and in S4 is the one-tailed probability. Cell S5 contains the absolute value of the test statistic, S6 contains the count of arrangement statistics that are at least as large in absolute value, and S7 contains the two-tailed probability.

Randomization Test Results for Design 2 Example

Row	*Result*	*1st run Col.S*	*2nd run Col.S*	*3rd run Col.S*
2	One-tailed statistic	2.592	2.592	2.592
3	No. as large	38	49	35
4	One-tailed probability	0.039	0.050	0.036

5	Two-tailed statistic	2.592	2.592	2.592
6	No. as large	38	49	35
7	Two-tailed probability	0.039	0.050	0.036

Mean time for three runs=4 min, 25 sec

Statistical Conclusion for Design 2 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 1000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.039. Therefore, the obtained difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 3 (AB MULTIPLE BASELINE)

Specifications for Design 3 Example

Number of participant (or behavior) replications (i.e., baselines)	=	3
Total number of observation periods per participant (or behavior)	=	18
Minimum number of control periods before treatment starts	=	6
Minimum number of treatment periods	=	6

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention points for the example are 9, 12, and 8.

Commented Macro for Design 3 (Macro File Name: design3.xls)

```
Sub design3()                                  clear area to be used
Columns("F:R").ClearContents                   calculate last possible
Range("A6").Select                             intervention point
ActiveCell.FormulaR1C1="=R3C1−R5C1+1"
Range("A7"). Select                            calculate number of available
                                               intervention points
ActiveCell.FormulaR1C1="=R6C1−R4C1"
Dim lastsubject As Integer                     number of participants
lastsubject=Range("A2")                        row number for bottom of
Dim last_i As Integer                          column of data and random
last_i=Range("A3")                             intervention codes for all
Dim lastrow$                                   participants
lastrow$=lastsubject * last_i+1
```

```
Dim lastperm$                          row number for bottom of
lastperm$=Range("A7")+1                columns of random
Dim lastphase$                         intervention points
lastphase$=last_i+1                    row number for bottom of
Range("F2").Select                     column of random intervention
                                       codes for one participant
                                       list the observation numbers
                                       from 1 to "A2"*"A3"

ActiveCell.FormulaR1C1="1"
Range("F3").Select
ActiveCell.FormulaR1C1="2"
Range("F2:F3").Select
Selection.AutoFill Destination:=Range("F2:F" & lastrow$), _
Type:=xlFillDefault                    fill in a list of cumulative
Range("G2").Select                     probabilities for the permitted
                                       intervention points

ActiveCell.FormulaR1C1="=(RC[-1]-1)/R7C1"
Selection.AutoFill Destination:=Range("G2:G" & lastperm$), _
Type:=xlFillDefault                    list the permitted intervention
Range("H2").Select                     points

ActiveCell.FormulaR1C1="=RC[-2]+R4C1"
Selection.AutoFill Destination:=Range("H2:H" & lastperm$), _
Type:=xlFillDefault                    make a lookup table with the
Columns("G:H").Select                  cumulative probabilities and the
                                       permitted intervention points

ActiveWorkbook.Names.Add Name:="lookup", RefersToR1C1:="=C7:C8"
Dim j As Integer                       number of randomly generated
Dim s As Integer                       intervention points per
For j=1 To 1000                        participant
Columns("L").CIearContents             this column will be used to store
For s=1 To lastsubject                 the intervention codes for all
Range("l2").Select                     participants at each arrangement:
                                       if not cleared at each
                                       arrangement it will keep
                                       accumulating the codes of all
                                       arrangements
                                       number of participants (each
                                       must be dealt with at every
                                       arrangement)
                                       find a random number and fill
                                       a column with it

ActiveCell.FormulaR1C1="=RAND()"
Selection.Copy
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _
SkipBlanks:=False, Transposes False
Selection.AutoFill Destination:=Range("l2:l" & lastperm$), _
```

```
Type:=xlFillDefault                          look up the intervention point
Range("J2").Select                           for this random number
ActiveCell.FormulaR1C1="=VLOOKUP(RC[−1], lookup,2)"
Range("K2").Select                           make a column of zeros and ones
                                             corresponding to this
                                             intervention point
ActiveCell.FormulaR1C1="=IF(R2C10>RC[−5],0,1)"
Selection.AutoFill Destination:=Range("K2:K" & lastphase$), _
Type:=xlFillDefault                          and store it before moving to the
Range("K2:K" & lastphase$).Select            next participant
Selection.Copy
Range("L2").Select
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                         next participant
Range("M2").Select                           make a column with −1 for the
                                             first preintervention row and a
                                             +1 for the first postintervention
                                             row for each participant for this
                                             arrangement
ActiveCell.FormulaR1C1=−1
Range("M3"). Select
ActiveCell.FormulaR1C1="=RC[−1]−R[−1]C[−1]"
Selection.AutoFill Destination:=Range("M3:M" & lastrow$), _
Type:=xlFillDefault
Range("E2").Select                           make a code that combines
                                             participant number and
                                             intervention code
ActiveCell.FormulaR1C1="=10*RC[−1]+RC[7]"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$), _
Type:=xlFillDefault
Range("N2").Select                           and use it to find the pre- and
                                             postrandom intervention means
                                             for each participant for this
                                             arrangement
ActiveCell.FormulaR1C1 C1="=SUMIF(level,C5,data)/COUNTIF(level,C5)"
Selection.AutoFill Destination:=Range("N2:N" & lastrow$), _
Type:=xlFillDefault
Range("O2").Select                           now use the column with −1 and
                                             +1 to get the difference between
                                             pre- and postrandom
                                             intervention means for this
                                             arrangement (when this column is
                                             summed)
```

```
ActiveCell.FormulaR1C1="=RC[−1]*RC[−2]"
Selection.AutoFill Destination:=Range("O2:O" & lastrow$),_
Type:=xlFillDefault
Range("P2").Select                          here is the arrangement test
                                            statistic for this arrangement
ActiveCell.FormulaR1C1="=SUM(C[−            store it
1])"
Selection.Copy
Range("Q2").Select
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                        next arrangement
lastj$=j+1                                  last row of arrangement
Range("M2").Select                          statistics
                                            now we need the column of -1s and+1s for
                                            the actual data
ActiveCell.FormulaR1C1=−1
Range("M3"). Select
ActiveCell.FormulaR1C1="=RC[−10]−R[−1]C[−10]"
Selection.AutoFill Destination:=Range("M3:M" & lastrow$), _
Type:=xlFillDefault
Range("E2").Select                          and the code combining
                                            participant and intervention code
                                            for the actual data
ActiveCell.FormulaR1C1="=10*RC[−1]+RC[−2]"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$),_
Type:=xlFillDefault                         and the pre- and postintervention
Range("N2").Select                          means for the actual data
ActiveCell.FormulaR1C1="=SUMIF(level,C5,data)/COUNTIF(level, C5)"
Selection.AutoFill Destination:=Range("N2:N" & lastrow$),_
Type:=xlFillDefault
Range("O2").Select                          now use the column with −1 and
                                            +1 to get the difference between
                                            pre- and postintervention means
                                            for the actual data (when this
                                            column is summed)
ActiveCell.FormulaR1C1="=RC[−1]*RC[−2]"
Selection.AutoFill Destination:=Range("O2:O" & lastrow$), _
Type:=xlFillDefault                         absolute value of the
Range("R2").Select                          arrangement statistics
ActiveCell.FormulaR1C1="=ABS(C[−1])"
Selection.AutoFill Destination:=Range("R2:R" & lastj$), _
Type:=xlFillDefault
```

```
Range("S2").Select                          here is the actual test statistic
ActiveCell.FormulaR1C1="=SUM(C[-            and the count of arrangement
4])"
Range("S3").Select

                                            statistics at least as great
ActiveCell.FormulaR1C1="=COUNTIF(C[-2]:C[-2],""">=""&R[-1]C)"
Range("S4").Select                          and the one-tailed probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("S5").Select                          the absolute value of the actual
                                            test statistic
ActiveCell.FormulaR1C1="=ABS(R[-3]C)"
Range("S6").Select                          and the count of arrangement
                                            statistics at least as large in
                                            absolute value as the absolute
                                            value of the actual test statistic
ActiveCell.FormulaR1C1="=COUNTIF(C[-1]:C[-1],""">=""&R[-1]C)"
Range("S7").Select                          and the two-tailed probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
End Sub
```

Location of Design 3 Test Results

After running, the test statistic (for the actual data) is in Cell S2. In S3 is the count of arrangement test statistics at least as large and in S4 is the one-tailed probability. Cell S5 contains the absolute value of the test statistic, S6 contains the count of arrangement statistics that are at least as large in absolute value, and S7 contains the two-tailed probability.

Randomization Test Results for Design 3 Example

Row	*Result*	*1st run Col.S*	*2nd run Col.S*	*3rd run Col.S*
2	One-tailed statistic	5.915	5.915	5.915
3	No. as large	35	51	33
4	One-tailed probability	0.036	0.052	0.034
5	Two-tailed statistic	5.915	5.915	5.915
6	No. as large	35	51	33
7	Two-tailed probability	0.036	0.052	0.034

Mean time for three runs=33 min, 40 sec

Statistical Conclusion for Design 3 (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of three communication aid users would increase when a word prediction system was introduced,

the proportion of 1000 randomly sampled data divisions giving a combined rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.036. Therefore, the obtained summed difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 4 (ABA MULTIPLE BASELINE DESIGN)

Specifications for Design 4 Example

Number of participants	=	2
Total number of observation periods per participant	=	12
Minimum number of control periods before treatment starts	=	3
Minimum number of treatment periods	=	3
Minimum number of control periods after withdrawal of treatment	=	3

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pairs of intervention and withdrawal points for the example are as follows: Participant 1 at Periods 6 and 10; Participant 2 at Periods 4 and 8.

Commented Macro for Design 4 (Macro File Name: design4.xls)

```
Sub design4()                                   clear area to be used
Columns("F:V").ClearContents                    calculate number of possible
Range("A7").Select                              intervention points
ActiveCell.FormulaR1C1="=(R3C1−R4C1−R5C1−R6C1+1)"
Range("A8").Select                              calculate number of possible
                                                intervention and withdrawal
                                                pairs
ActiveCell.FormulaR1C1="=R7C1*(R7C1+1)/2"
Dim lastsubject As Integer                      number of participants
lastsubject=Range("A2")                         row number for bottom of
Dim last_i As Integer                           column of data and random
last_i=Range("A3")                              intervention codes for all
Dim lastrow$                                    participants
lastrow$=lastsubject * last_i+1                 row number for bottom of
Dim lastperm$                                   columns of random
                                                intervention and withdrawal
                                                pairs
lastperm$=Range("A8")+1                         row number for bottom of
Dim lastphase$                                  columns of random
                                                intervention codes for one
                                                participant
```

```
lastphase$=last_i+1                          list the observation numbers
Range("F2").Select                           from 1 to "A2"*"A3"
ActiveCell.FormulaR1C1="1"
Range("F3").Select
ActiveCell.FormulaR1C1="2"
Range("F2:F3").Select
Selection.AutoFill Destination:=Range("F2:F" & lastphase$), _
Type:=xlFillDefault
Range("G2").Select                           columns G, H, I enable us to list
                                             the possible intervention and
                                             withdrawal pairs in columns K
                                             and L
ActiveCell.FormulaR1C1="=INT(0.5+SQRT(2*RC[-1]))"
Selection.AutoFill Destination:=Range("G2:G" & lastperm$), _
Type:=xlFillDefault
Range("H2").Select
ActiveCell.FormulaR1C1="=INT((SQRT(8*(RC[-2]-1)+1)-1)/2)"
Selection.AutoFill Destination:=Range("H2:H" & lastperm$), _
Type:=xlFillDefault
Range("I2").Select
ActiveCell.FormulaR1C1="=RC[-3]-RC[-1]*(RC[-1]+1)/2"
Selection.AutoFill Destination:=Range("I2:I" & lastperm$), _
Type:=xlFillDefault
Range("K2").Select
ActiveCell.FormulaR1C1="=R3C1-R5C1-R6C1+1-RC[-3]"
Selection.AutoFill Destination:=Range("K2:K" & lastperm$), _
Type:=xlFillDefault
Range("L2").Select
ActiveCell.FormulaR1C1="=R3C1-R6C1+2-RC[-3]"
Selection.AutoFill Destination:=Range("L2:L" & lastperm$), _
Type:=xlFillDefault
Range("J2").Select                           column J has the cumulative
ActiveCell.FormulaR1C1="=(RC[-4]-1)/R8C1"    probabilities for the possible
                                             pairs
Selection.AutoFill Destination:=Range("J2:J" & lastperm$), _
Type:=xlFillDefault
Columns("J:L").Select                        make a lookup table with the
                                             cumulative probabilities and the
                                             permitted intervention and
                                             withdrawal points
ActiveWorkbook.Names.Add Name:="lookup" RefersToR1C1:="=C10:C12"
Dim j As Integer
Dim s As Integer
```

```
For j=1 To 1000                                  number of randomly generated
                                                 intervention and withdrawal
                                                 pairs per participant
Columns("P").ClearContents                       this column will be used to store
                                                 the intervention codes for all
                                                 participants at each arrangement
                                                 if not cleared at each
                                                 arrangement it will keep
                                                 accumulating the codes of all
                                                 arrangements
For s=1 To lastsubject                           number of participants (each
                                                 must be dealt with at every
                                                 arrangement)
Range("M2").Select                               find a random number and fill
                                                 a column with it
ActiveCell.FormulaR1C1="=RAND()"
Selection.Copy
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, _
SkipBlanks:=False, Transposes False
Selection.AutoFill Destination:=Range("M2:M" & lastperm$), _
Type:=xlFillDefault
Range("N2").Select                               look up the intervention point
                                                 for this random number
ActiveCell.FormulaR1C1="=VLOOKUP(RC[-1],lookup,2)"
Range("N3").Select                               and the withdrawal point
ActiveCell.FormulaR1C1="=VLOOKUP(RC[-1],lookup,3)"
Range("O2").Select                               make a column of zeros and
                                                 ones corresponding to this
                                                 intervention and withdrawal
                                                 pair
ActiveCell.FormulaR1C1="=IF(RC[-9]<R2C14, 0, IF(RC[-9]>=R3C14, 0, 1))"
Selection.AutoFill Destination:=Range("O2:O" & lastphase$), _
Type:=xlFillDefault
Range("O2:O" & lastphase$).Select
Selection.Copy
Range("P2").Select                               and store it before moving to the
Selection.Insert Shift:=xlDown                   next participant
Selection.PasteSpecial Paste:=xlValues
Next                                             next participant
Range("Q2").Select                               make a column with+1 for the
                                                 first intervention row and a -1
                                                 for the first withdrawal row for
                                                 each participant for this
                                                 arrangement
```

```
ActiveCell.FormulaR1C1=0
Range("Q3").Select
ActiveCell.FormulaR1C1\C1="=RC[−1]−R[−1]C[−1]"
Selection.AutoFill Destination:=Range("Q3:Q" & lastrow$), _
Type:=xlFillDefault
Range("E2").Select                          make a code that combines
                                            participant number and
                                            intervention code
ActiveCell.FormulaR1C1="=10*RC[−1]+RC[11]"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$), _
Type:=xlFillDefault
Range("R2").Select                          and use it to find the pre-, post-,
                                            and during random intervention
                                            means for each participant for
                                            this arrangement
ActiveCell.FormulaR1C1="=SUMIF(level, C5,data)/COUNTIF(level, C5)"

Selection.AutoFill Destination:=Range("R2:R" & lastrow$), _
Type:=xlFillDefault                         now use the column with −1 and
Range("S2").Select                          +1 to get the difference between
                                            pre-, post-, and during random
                                            intervention means for this
                                            arrangement (when this column is
                                            summed)
ActiveCell.FormulaR1C1="=RC[−1]*RC[−2]"
Selection.AutoFill Destination:=Range("S2:S" & lastrow$), _
Type:=xlFillDefault
Range("T2").Select                          here is the arrangement test
                                            statistic for this arrangement
ActiveCell.FormulaR1C1="=SUM(C[−1])"
Selection.Copy
Range("U2").Select                          store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                        and go on to the next arrangement
lastj$=j+1                                  last row of arrangement
Range("Q2").Select                          statistics
                                            now we need the column of-1s
                                            and+1s for the actual data
ActiveCell.FormulaR1C1=0
Range("Q3").Select
ActiveCell.FormulaR1C1="=RC[−14]−R[−1]C[−14"
Selection.AutoFill Destination:=Range("Q3:Q" & lastrow$), _
Type:=xlFillDefault
```

```
Range("E2").Select                                  and the code combining
                                                    participant and intervention code
                                                    for the actual data
ActiveCell.FormulaR1C1="=10*RC[-1]+RC[-2]"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$), _
Type:=xlFillDefault
Range("R2").Select                                  and the pre-, post-, and during
                                                    intervention means for the
                                                    actual data
ActiveCell.FormulaR1C1="=SUMIF(level,C5,data)/COUNTIF(level,C5)"
Selection.AutoFill Destination:=Range("R2:R" & lastrow$), _
Type:=xlFillDefault
Range("S2").Select                                  now use the column with -1 and
                                                    +1 to get the difference between
                                                    pre-, post-, and during intervention
                                                    means for the actual data (when
                                                    this column is summed)
ActiveCell.FormulaR1C1="=RC[-1]*RC[-2]"
Selection.AutoFill Destination:=Range("S2:S" & lastrow$),_
Type:=xlFillDefault
Range("V2").Select                                  absolute value of arrangement
                                                    statistics
ActiveCell.FormulaR1C1="=ABS(C[-1])"
Selection.AutoFill Destination:=Range("V2:V" & lastj$), _
Type:=xlFillDefault
Range("W2").Select                                  here is the actual test statistic
ActiveCell.FormulaR1C1="=SUM(C[-4])"
Range("W3").Select
                                                    and the count of arrangement
                                                    statistics at least as great
ActiveCell.FormulaR1C1="=COUNTIF(C[-2]:C[-2],""">=""&R[-1]C)"
Range("W4").Select                                  and the one-tailed probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("W5").Select                                  absolute value of actual test
                                                    statistic
ActiveCell.FormulaR1C1="=ABS(R[-3]C)"
Range("W6").Select
                                                    and the count of arrangement
                                                    statistics at least as great in
                                                    absolute value as the absolute
                                                    value of the actual test statistic
ActiveCell.FormulaR1C1="=COUNTIF(C[-1]:C[-1],""">=""&R[-1]C)"
Range("W7").Select                                  and the two-tailed probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
End Sub
```

Location of Design 4 Test Results

After running, the test statistic (for the actual data) is in Cell W2. In W3 is the count of arrangement test statistics at least as large and in W4 is the one-tailed probability. Cell W5 contains the absolute value of the test statistic, W6 contains the count of arrangement statistics that are at least as large in absolute value, and W7 contains the two-tailed probability.

Randomization Test Results for Design 4 Example

Row	*Result*	*1st run Col.W*	*2nd run Col.W*	*3rd run Col.W*
2	One-tailed statistic	3.750	3.750	3.750
3	No. as large	32	27	25
4	One-tailed probability	0.033	0.028	0.026
5	Two-tailed statistic	3.750	3.750	3.750
6	No. as large	32	27	25
7	Two-tailed probability	0.033	0.028	0.026

Mean time for three runs=18 min, 10 sec

Statistical Conclusion for Design 4 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of two communication aid users would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 1000 randomly sampled data divisions giving a combined rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.033. Therefore, the obtained summed difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 5 (ONE-WAY SMALL GROUPS AND SINGLE-CASE RANDOMIZED TREATMENT)

Specifications for Design 5 Example

Total number of participants (or treatment occasions)	=	10
Number of treatments: Communication aid systems (or levels of symbol translucency)	=	4
Number of participants per communication system (or symbol sets per translucency level)	=	2, 2, 3, 3

Commented Macro for Design 5 (Macro File Name: design5.xls)

```
Sub design5()
Columns("E:O").CIearContents                    clear area to be used
                                                for counting arrangements
Dim j As Integer
Dim i As Integer
Dim last_i As Integer                           number of observations
last_i=Range("A2")
Dim lastrow$                                    row number for bottom of
                                                columns of data and
                                                arrangements
lastrow$=last_i+1
Range("B2:B" & lastrow$).Select                 copy the observations so they
Selection.Copy                                  can be arranged
Range("D2").Select
ActiveSheet.Paste
For j=1 To 1000                                 number of arrangements
Range("E2").Select                              put a column of random
                                                numbers next to the copy of the
                                                observations
ActiveCell.FormulaR1C1="=Rand()"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$), _
Type:=xlFillDefault
Range("D2:E" & lastrow$).Select                 and sort the random numbers
                                                into order, carrying the copy of
                                                the observations so we have an
                                                arrangement
Selection.Sort Key1:=Range("E2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatehCase:=False, _
Orientation:=xlTopTo Bottom
Range("F2").Select
ActiveCell.FormulaR1C1=_
"=Sumif(level,RC[−3],arrange)/Countif(level,RC[−3])"
                                                fill in the treatment group means
                                                for the arrangement (these are
                                                the fitted values)
Selection.AutoFill Destination:=Range("F2:F" & lastrow$), _
Type:=xlFillDefault
Range("G2").Select
ActiveCell.FormulaR1C1="=RC[−3]−RC[−1]"         fill in the residuals for the
                                                arrangement
Selection.AutoFill Destination:=Range("G2:G" & lastrow$), _
Type:=xlFillDefault
Range("H2").Select
```

```
ActiveCell.FormulaR1C1="=Sumsq(C[−1])"          RSS for the arrangement
Selection.Copy
Range("l2").Select
Selection.Insert Shift:=xlDown                  store the RSS from each
Selection.PasteSpecial Paste:=xlValues          arrangement
Next
                                                next arrangement
Range("J2").Select
ActiveCell.FormulaR1C1=_
"=Sumif(level,RC[−7],data)/Countif(level,RC[−7])"
                                                fill in the treatment group means
                                                for the actual experiment
Selection.AutoFill Destination:=Range("J2:J" & lastrow$), _
Type:=xlFillDefault
Range("K2").Select
ActiveCell.FormulaR1C1="=RC[−9]−RC[−1"
                                                fill in the actual residuals
Selection.AutoFill Destination:=Range("K2:K" & lastrow$), _
Type:=xlFillDefault
Range("L2").FormulaR1C1="=Sumsq(C[−1])"         actual RSS
Range("L3").FormulaR1C1=_                       count the cases where the
"=Countif(C[−3]:C[−3],""<="""&R[−1]C)"          arrangement RSS is ≤ actual
Range("L4").FormulaR1C1="=(R[−1]C+1)/(1000+1)"
End Sub                                         probability of a result at least as
                                                extreme as the actual one
```

Location of Design 5 Test Results

After running, the test statistic (for the actual data) is in Cell L2. In L3 is the count of arrangement test statistics at least as small (because the statistic is the RSS) and in L4 is the two-tailed probability.

Randomization Test Results for Design 5 Example

Row	*Result*	*1st run Col.L*	*2nd run Col.L*	*3rd run Col.L*
2	Two-tailed statistic	3.333	3.333	3.333
3	No. as small	7	4	5
4	Two-tailed probability	0.008	0.005	0.006

Mean time for three runs=1 min, 28 sec

Statistical Conclusion for Design 5 (One-Way Small Groups) Example

In a randomization test of the prediction that four communication aids would differ in the time taken to learn to use them to a criterion, the proportion of 1000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.008. Therefore, the obtained differences in learning time with the four communication aids were statistically significant ($p<0.01$).

DESIGN 5a (SMALL GROUPS OR SINGLE-CASE—TWO RANDOMIZED TREATMENTS)

Specifications for Design 5a Example

Total number of participants (or treatment occasions)	=	9
Number of treatments: Communication aid systems (or levels of symbol translucency)	=	2
Number of participants per communication system (or symbol sets per translucency level)	=	4, 5

For a one-tailed test the level with the higher expected mean is coded 2 and the level with the lower expected mean is coded 1.

Commented Macro for Design 5a (Macro File Name: design5a.xls)

```
Sub design5a()
Columns("E:O") .ClearContents      clear area to be used
Dim j As Integer                   for counting arrangements
Dim i As Integer
Dim last_i As Integer
last_i=Range("A2")                 number of observations
Dim lastrow$
                                   row number for bottom of
                                   columns of data and
                                   arrangements
lastrow$=last_i+1
Range("B2:B" & lastrow$).Select
Selection.Copy                     copy the observations so they
                                   can be rearranged
Range("D2").Select
ActiveSheet.Paste
For j=1 To 1000                    number of arrangements
Range("E2").Select                 put a column of random
                                   numbers next to the copy of the
                                   observations
ActiveCell.FormulaR1C1="=Rand()"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$), _
```

```
Type:=xlFillDefault
Range("D2:E" & lastrow$).Select          and sort the random numbers into order, carrying the copy of the observations so we have an arrangement
Selection.Sort Key1:=Range("E2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopToBottom
Range("F2").Select                       find the mean of the observations at level 2
ActiveCell.FormulaR1C1=_
"=Sumif(C[−3]:C[−3],""=2"",C[−2]:C[−2])/Countif(C[−3]:C[−3],""=2"")"
Range("F3").Select                       and the mean at level 1
ActiveCell.FormulaR1C1=_
"=Sumif(C[−3]:C[−3],""=1"",C[−2]:C[−2])/Countif(C[−3]:C[−3],""=1"")"
Range("F4").Select
                                         find the difference between the level 2 and level 1 means
ActiveCell.FormulaR1C1="=R[−2]C−R[−1]C"
Selection.Copy
Range("G2").Select                       and store the value for this arrangement
Selection.Insert Shift:=xlDown
Selection.PasteSpecial
Paste:=xlValues
Range("F5").Select                       find the absolute value of the difference for the two-tailed test
ActiveCell.
FormulaR1C1="=ABS(R[−1]C)"
Selection.Copy
Range("H2").Select                       and store it for this arrangement
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                     next arrangement
Range("F6").Select                       now find the level 2 mean for the actual observations
ActiveCell.FormulaR1C1=_
"=Sumif(C[−3]:C[−3],""=2"",C[−4]:C[−4])/Countif(C[−3]:C[−3],""=2"")"
Range("F7").Select                       and the level 1 mean for the actual observations
ActiveCell.FormulaR1C1=_
"=Sumif(C[−3]:C[−3],""=1"",C[−4]:C[−4])/Countif(C[−3]:C[−3],""=1"")"
```

```
Range("F8").Select                          and the difference
ActiveCell.FormulaR1C1="=R[-2]
C-R[-1]C"
Range("F9").Select                          and its absolute value
ActiveCell.
FormulaR1C1="=ABS(R[-1]C)"                  put the test statistic in 12
Range("l2").FormulaR1C1="=R[6]
C[-3]"
Range("13").FormulaR1C1=_                   count the cases for the one-tailed
"=Countif(C[-2]:C[-2],"">=""&R              test, then find the probability of
[5]C[-3])"                                  your value or higher
Range("l4").FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("l5").FormulaR1C1="=R[4]C[-3]"
Range("l6").FormulaR1C1=_
"=Countif(C[-1]:C[-1],"">=""&R              count the cases for the two-tailed
[3]C[-3])"                                  test, then find the probability of
                                            your value or higher
Range("17").FormulaR1C1="=(R[-1]C+1)/(1000+1)"
End Sub
```

Location of Design 5a Test Results

After running, the test statistic (for the actual data) is in Cell 12. In 13 is the count of arrangement test statistics at least as large and in 14 is the one-tailed probability. Cell 15 contains the absolute value of the test statistic, 16 contains the count of arrangement statistics that are at least as large in absolute value, and 17 contains the two-tailed probability.

Randomization Test Results for Design 5a Example

Row	*Result*	*1st run Col. I*	*2nd run Col.*	*3rd run Col.*
2	One-tailed statistic	1.550	1.550	1.550
3	No. as large	40	44	31
4	One-tailed probability	0.041	0.045	0.032
5	Two-tailed statistic	1.550	1.550	1.550
6	No. as large	64	82	56
7	Two-tailed probability	0.065	0.083	0.057

Mean time for three runs=1 min, 50 sec

Statistical Conclusion for Design 5a (One-Tailed Single-Case) Example

In a randomization test of the prediction that high-translucency symbols will take fewer sessions than low-translucency symbols for a communication aid user to learn, the proportion of 1000 randomly sampled data divisions giving a learning sessions difference in the

predicted direction at least as large as the experimentally obtained difference was 0.041. Therefore, the obtained difference in learning sessions required was statistically significant ($p<0.05$; one-tailed).

It may be noted that, had a directional prediction not been made in this instance, the two-tailed test would not have shown a significant difference ($p=0.065$).

DESIGN 6 (ONE-WAY SMALL GROUP REPEATED MEASURES AND SINGLE-CASE RANDOMIZED BLOCKS)

Specifications for Design 6 Example

Number of participants (or number of blocks)	=	3
Number of conditions	=	4
Total number of observations (conditions×participants or blocks)	=	12

Each participant must receive the same number of measures (or each treatment must appear once in each block).

Commented Macro for Design 6 (Macro File Name: design6.xls)

```
Sub design6()
Columns("E:Q").ClearContents                    clear cells to be used
Dim j As Integer                                this will count arrangements
Dim lastobs$                                    number of observations
lastobs=Range("A2")
Dim lastrow$
lastrow$=lastobs+1                              this is the last row of data and
Range("B2:B" & lastrow$).Select                 associated columns
Selection.Copy
Range("E2"). Select                             put a copy of the observations
ActiveSheet. Paste                              in column E to be arranged
For j=1 To 1000
Range("F2").Select
                                                number of arrangements
                                                fill a column with random
                                                numbers and add the block
                                                number to so that arrangements
                                                are within blocks
ActiveCell.FormulaR1C1="=RAND()+RC[−2]"
Selection.AutoFill Destination:=Range("F2:F" & lastrow$), _
Type:=xlFillDefault
Range("E2:F" & lastrow$).Select                 sort random number +block
                                                column and carry along the copy
                                                of observations to give an
                                                arrangement within blocks
```

```
Selection.Sort Key1:=Range("F2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, Match Case:=False, _
Orientation:=xlTopToBottom
Range("G2").Select                                  find the treatment totals for this
                                                    arrangement
ActiveCell.FormulaR1C1=_
"=SUMIF(condition,RC[-4],perm)"
Selection.AutoFill Destination:=Range("G2:G" & lastrow$)
Range("H2").Select                                  and square them
ActiveCell.FormulaR1C1="=RC[-1]*RC[-1]"
Selection.AutoFill Destination:=Range("H2:H" & lastrow$)
Range("l2").Select                                  find the block totals for this
                                                    arrangement
ActiveCell.FormulaR1C1=_
"=SUMIF(block,RC[-5],perm)"
Selection.AutoFill Destination:=Range("l2:l" & lastrow$)
Range("J2").Select                                  and square them
ActiveCell.FormulaR1C1="=RC[-1]*RC[-1]"
Selection.AutoFill Destination:=Range("J2:J" & lastrow$)
Range("K2").Select                                  find the RSS for this
                                                    arrangement (don't forget H
                                                    contains a copy for every block
                                                    and J a copy for every
                                                    treatment)
ActiveCell.FormulaR1C1=_
"=SUMSQ(C2)-SUM(C[-3])/R3C1^2-SUM(C[-1])/(R2C1/R3C1)^2+SUM(data)^2/R2C1"
Selection.Copy
Range("L2").Select                                  and store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste: =xlValues
Next                                                next arrangement
Range("M2").Select                                  find the treatment totals for the
ActiveCell.FormulaR1C1=_                            actual data
"=SUMIF(condition, RC[-10],data)"
Selection.AutoFill Destination:=Range ("M2:M" & lastrow$)
Range("N2").Select                                  and square them
ActiveCell.FormulaR1C1="=RC[-1]*RC[-1]"
Selection.AutoFill Destination:=Range("N2:N" & lastrow$)
Range("O2").Select                                  find the block totals for the actual data
ActiveCell.FormulaR1C1=_
"=SUMIF(block,RC[-11],data)"
Selection.AutoFill Destination:=Range("O2:O" & lastrow$)
Range("P2").Select                                  and square them
```

```
ActiveCell.FormulaR1C1="=RC[−1]*RC[−1]"
Selection.AutoFill Destination:=Range("P2:P" & lastrow$)
Range("Q2").Select                                  find the RSS for the actual data
ActiveCell.FormulaR1C1=
"=SUMSQ(C2)−SUM(C[−3])/R3C1^2−SUM(C[−1])/(R2C1/R3C1)^2+SUM(data)^2/R2C
Range("Q3").Select                                  count the arrangements with a
                                                    test statistic at least as small
ActiveCell.FormulaR1C1="=COUNTIF(C[−5]:C[−5],""<="""&R[−1]C)"
Range("Q4").Select                                  and find the probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
End Sub
```

Location of Design 6 Test Results

After running, the test statistic (RSS for the actual data) is in Cell Q2. In Q3 is the count of arrangement test statistics at least as small (because the statistic is the RSS) and in Q4 is the two-tailed probability.

Randomization Test Results for Design 6 Example

Row	*Result*	*1st run Col.Q*	*2nd run Col.Q*	*3rd run Col.Q*
2	Two-tailed statistic	98.500	98.500	98.500
3	No. as small	3	0	2
4	Two-tailed probability	0.004	0.001	0.003

Mean time for three runs=2 min, 46 sec

Statistical Conclusion for Design 6 (One-Way Small Groups) Example

In a randomization test of the prediction that the number of initiations made by communication aid users will differ depending on the level of experience of conversational partners, the proportion of 1000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.004. Therefore, the obtained differences in number of user initiations with partners with different levels of experience was statistically significant ($p<0.01$).

DESIGN 6a (TWO REPEATED MEASURES ON SMALL GROUP OR SINGLE-CASE BLOCKS)

Specifications for Design 6a Example

Number of participants (or number of blocks)	=	7
Number of conditions	=	2

Each participant must receive the same number of measures (or each treatment must appear once in each block).

For a one-tailed test the condition with the higher expected mean is coded 2 and the condition with the lower expected mean is coded 1.

Commented Macro for Design 6a (Macro File Name: design6a.xls)

```
Sub design6a()
Columns("H:M").CIearContents                    clear the columns to be used
                                                except for E to G which will be
                                                cleared at each arrangement
Dim j As Integer
Dim lastblock As Integer
lastblock=Range("A2")
Dim lastrow$                                    this will count arrangements
                                                we need to record number
                                                of participants
                                                this is the last row of data and
                                                associated columns
lastrow$=2 * lastblock+1
For j=1 To 1000                                 number of arrangements
Columns("E:G").CIearContents                    clear columns for random
                                                numbers, arrangement of
                                                conditions and data for
                                                condition 2 in this arrangement
Range("E2").Select
ActiveCell.FormulaR1C1="=RAND()"                fill a column with random
                                                numbers (alternate ones only
                                                will be used)
Selection.AutoFill Destination:=Range("E2:E" & lastrow$)
Range("F2").Select
ActiveCell.FormulaR1C1="=IF(RC[−1]>0.5, 0, 1)"  for each participant make the
                                                first observation condition 1 if
                                                the random number is>0.5,
                                                otherwise condition 2
Range("F3"). Select
ActiveCell.FormulaR1C1="=1−R[−1]C"              the second observation for each
Range("F2:F3").Select                           participant is assigned whichever
                                                condition the first was not
                                                fill the column so this is done
                                                for each participant
Selection.AutoFill Destination:=Range("F2:F" & lastrow$)
Range("G2"). Select
ActiveCell.FormulaR1C1="=RC[−5]*RC[−1]"         multiply the data by zero
                                                (condition 1) or 1 (condition 2)
                                                for this arrangement
Selection.AutoFill Destination:=Range("G2:G" & lastrow$)
```

```
Range("H2").Select                                  calculate the test statistic
                                                    (difference between condition
                                                    totals) for this arrangement
ActiveCell.FormulaR1C1="=2*SUM(C[−1]:C[−1])−SUM(C[−6]:C[−6])"
Selection.Copy                                      and store it
Range("l2"). Select
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                                next arrangement
lastperm$=j                                         we need to use the number of
                                                    arrangements in getting the
                                                    two-tailed probability
Range("J2").Select
ActiveCell.FormulaR1C1="=ABS(RC[−1])"               find absolute values for the
                                                    arrangement test statistics
Selection.AutoFill Destination:=Range("J2:J" & lastperm$)
Range("K2").Select
ActiveCell.FormulaR1C1="=RC[−9]*(RC[−8]−1)"
Selection.AutoFill Destination:=Range("K2:K" & lastrow$)
Range("L2").Select                                  find the test statistic for the
                                                    actual data
ActiveCell.FormulaR1C1="=2*SUM(C[−1]:C[−1])−SUM(C[−10]:C[−10])"
Range("L3").Select                                  and its absolute value
ActiveCell.FormulaR1C1="=ABS(R[−1]C"
Range("M2").Select
                                                    the difference between condition
                                                    means may be preferred to the
                                                      (equivalent) difference between
                                                      totals
ActiveCell.FormulaR1C1="=RC[−1]/R2C1"
Range("M3").Select                                  count the arrangement statistics
                                                    that are at least as great as the
                                                    one from the actual data
ActiveCell.FormulaR1C1="=COUNTIF(C[−4]:C[−4],""">=""&R[−1]C[−1])"
Range("M4").Select                                  and get the one-tailed
                                                    probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
Range("M5").Select                                  absolute difference between
                                                    means for actual data
ActiveCell.FormulaR1C1="=R[−2]C[−1]/R2C1"
Range("M6").Select                                  count of absolute values of
                                                    arrangement statistics that are
                                                    at least as great
ActiveCell.FormulaR1C1="=COUNTIF(C[−3]:C[−3],""">=""&R[−3]C[−1])"
Range("M7").Select                                  two-tailed probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1"
End Sub
```

Location of Design 6a Test Results

After running, the test statistic (for the actual data) is in Cell M2. In M3 is the count of arrangement test statistics at least as large and in M4 is the one-tailed probability. Cell M5 contains the absolute value of the test statistic, M6 contains the count of arrangement statistics that are at least as large in absolute value, and M7 contains the two-tailed probability.

Randomization Test Results for Design 6a Example

Row	*Result*	*1st run Col.M*	*2nd run Col.M*	*3rd run Col.M*
2	One-tailed statistic	1.857	1.857	1.857
3	No. as large	45	49	59
4	One-tailed probability	0.046	0.050	0.060
5	Two-tailed statistic	1.857	1.857	1.857
6	No. as large	90	89	100
7	Two-tailed probability	0.091	0.090	0.101

Mean time for three runs=1 min, 52 sec

Statistical Conclusion for Design 6a (One-Tailed Single-Case) Example

In a randomization test of the prediction that a communication aid user will choose to use a high-tech aid more frequently with a speech and language therapist than with a family member, the proportion of 1000 randomly sampled data divisions giving a difference in high-tech aid use in the predicted direction at least as large as the experimentally obtained difference was 0.046. Therefore, the obtained difference in high-tech use was statistically significant ($p<0.05$; one-tailed).

As in Design 5a, if a directional prediction had not been made in this instance, the two-tailed test would not have shown a significant difference ($p=0.091$).

DESIGN 7 (TWO-WAY FACTORIAL SINGLE CASE)

Specifications for Design 7 Example

Factor 1=interface device (touch-screen coded 1, joystick coded 2)
Factor 2=display mode (static coded 1, dynamic coded 2)

Total number of observations	=	16
Number of observations per condition (must be equal)	=	4

Directional prediction for Factor 1: joystick>touch-screen
Directional prediction for Factor 2: dynamic>static
Predicted interaction: Rate slowest for touch-screen interface with dynamic display (see Fig. 5.1)
Predictions for simple effects based on predicted interaction:
dynamic>static with touch-screen interface

joystick>touch-screen with static display
dynamic will not differ significantly from static with joystick interface
joystick will not differ significantly from touch-screen with dynamic display

Commented Macro for Design 7 (Macro File Name: design7.xls)

```
Sub design7()
Columns("E:U").CIearContents                    clear columns to be used
Dim j As Integer                                this will count arrangements
Dim lastobs$                                    number of observations
lastobs=Range("A2")
Dim lastrow$                                    this is the last row of data and
lastrow$=lastobs+1                              associated columns
Range("B2:D" & lastrow$).Select                 to deal with factor 1 we need
                                                the data ordered with the
                                                factor 2 levels in blocks
Selection.Sort Key1:=Range("D2"), Order1:=xlAscending, _
Key2:=Range("C2"), Order2:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopTo Bottom
Range("B2:B" & lastrow$).Select
Selection.Copy                                  make a copy of the correctly
Range("E2").Select                              ordered data for arrangements
ActiveSheet. Paste                              within levels of factor 2
For j=1 To 1000
Range("F2").Select
                                                number of arrangements for factor 1
                                                fill a column with random
                                                numbers plus the factor 2 levels
ActiveCell.FormulaR1C1="=RAND()+RC[−2]"
Selection.AutoFill Destination:=Range("F2:F" & lastrow$), _
Type:=xlFillDefault
Range("E2:F" & lastrow$).Select                 and sort it, carrying along the
                                                data copy, to get an arrangement within factor
                                                2 levels
Selection.Sort Key1:=Range("F2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopToBottom
Range("G2").Select                              multiply the arranged
                                                observations by −1 for level 1 and
                                                +1 for level 2 of factor 1
ActiveCell.FormulaR1C1="=(RC[−2]*(RC[−4]−1.5)*2)"
Selection.AutoFill Destination:=Range("G2:G" & lastrow$)
Range("H2").Select                              and find the difference between
                                                factor level means for factor 1
```

```
ActiveCell.FormulaR1C1="=2*SUM(C[−1])/R2C1"
Selection.Copy
Range("l2").Select                          and store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial
Paste:=xlValues                             next arrangement within levels of
Next                                        factor 2
Range("B2:D" & lastrow$).Select             now we have to reorder the data so
                                            that factor 1 levels are in blocks
Selection.Sort Key1:=Range("C2"), Order1:=xlAscending, _
Key2:=Range("D2"), Order2:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopToBottom
Range("B2:B" & lastrow$).Select
Selection.Copy                              now make a copy of the
Range("J2").Select                          observations for arrangements
ActiveSheet. Paste                          within levels of factor 1
For j=1 To 1000
Range("K2").Select
                                            number of arrangements for factor
                                            fill a column with random
                                            numbers plus the factor 1 levels
ActiveCell.FormulaR1C1="=RAND()+RC[−8]"
Selection.AutoFill Destination:=Range("K2:K" & lastrow$), _
Type:=xlFillDefault
Range("J2:K" & lastrow$).Select             and sort it, carrying along the
                                            data copy, to get an arrangement
                                            within factor 1 levels
Selection.Sort Key1:=Range("K2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopToBottom
Range("L2").Select                          multiply the arranged
                                            observations by −1 for level 1
                                            and +1 for level 2 of factor 2
ActiveCell.FormulaR1C1="=(RC[−2]*(RC[−8]−1.5)*2)"
SelectJon.AutoFill Destination:=Range("L2:L" & lastrow$)
Range("M2").Select                          and find the difference between
                                            factor level means for factor 2
ActiveCell.FormulaR1C1="=2*SUM(C[−1])/R2C1"
Selection.Copy
```

```
Range("N2").Select                          and store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                        next arrangement for factor 2
lastj$=j+1                                  last row of arrangement statistics
Range("O2").Select
                                            find absolute values of arrangement test statistics for factor 1
ActiveCell.FormulaR1C1="=ABS(C[−6])"
Selection.AutoFill Destination:=Range("O2:O" & lastj$)
Range("P2").Select                          and factor 2
ActiveCell.FormulaR1C1="=ABS(C[−2])"
Selection.AutoFill Destination:=Range("P2:P" & lastj$)
Range("Q2").Select                          multiply the actual observations by −1 for level 1 and +1 for level 2 of factor 1
ActiveCell.FormulaR1C1="=(RC[−15]*(RC[−14]−1.5)*2)"
Selection.AutoFill Destination:=Range("Q2:Q" & lastrow$)
Range("R2").Select                          and find the difference between factor level means for factor 1
ActiveCell.FormulaR1C1="=2*SUM(C[−1])/R2C1"
Range("R3").Select                          and its absolute value
ActiveCell.
FormulaR1C1="=ABS(R[−1]C)"
Range("S2").Select                          multiply the actual observations by −1 for level 1 and +1 for level 2 of factor 2
ActiveCell.FormulaR1C1="=(RC[−17]*(RC[−15]−1.5)*2)"
Selection.AutoFill Destination:=Range("S2:S" & lastrow$)
Range("T2").Select                          and find the difference between factor level means for factor 2
ActiveCell.FormulaR1C1="=2*SUM(C[−1])/R2C1"
Range("T3").Select                          and its absolute value
ActiveCell.
FormulaR1C1="=ABS(R[−1]C)"                  one-tailed factor 1 test statistic
Range("U2").Select                          number of arrangement test statistics as least as large
ActiveCell.FormulaR1C1="=RC[−3]"
Range("U3").Select
ActiveCell.FormulaR1C1="=COUNTIF(C[−12]:C[−12],"">=""&R[−1]C)"
Range("U4").Select                          one-tailed factor 1 probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)"
Range("U5").Select                          two-tailed factor 1 test statistic
ActiveCell.FormulaR1C1="=R[−2]
C[−3]"
```

```
Range("U6").Select                          number of arrangement test
                                            statistics as least as large
ActiveCell.FormulaR1C1="=COUNTIF(C[-6]:C[-6],""">=""&R[-1]C)"
Range("U7").Select                          two-tailed factor 1 probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("U8").Select                          one-tailed factor 2 test statistic
ActiveCell.FormulaR1C1="=R[-6]
C[-1]"
Range("U9").Select                          number of arrangement test
                                            statistics as least as large
ActiveCell.FormulaR1C1="=COUNTIF(C[-7]:C[-7],""">=""&R[-1]C)"
Range("U10").Select                         one-tailed factor 2 probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
Range("U11").Select                         two-tailed factor 2 test statistic
ActiveCell.FormulaR1C1="=R[-8]
C[-1]"
Range("U12").Select                         number of arrangement test
                                            statistics as least as large
ActiveCell.FormulaR1C1="=COUNTIF(C[-5]:C[-5],""">=""&R[-1]C)"
Range("U13").Select                         two-tailed factor 2 probability
ActiveCell.FormulaR1C1="=(R[-1]C+1)/(1000+1)"
End Sub
```

Location of Design 7 Test Results

After running, Column U contains the results. Cell U2 is the one-tailed test statistic for Factor 1 (the mean of Level 2—the mean of Level 1). U3 contains the count of arrangement test statistics at least as large and in U4 is the one-tailed probability. U5 contains the absolute value of the test statistic, U6 contains the number of arrangement statistics that are at least as large in absolute value, and U7 contains the two-tailed probability. Cells U8 through U10 contain the one-tailed results for Factor 2 and Cells U11 through U13 contain the two-tailed results for Factor 2. To examine simple effects, use Design 5a. An example for one of the simple effects is provided later.

Randomization Test Results for Design 7 Example

Row	*Result*	*1st run Col. U*	*2nd run Col. U*	*3rd run Col.U*
	Main Effect for Factor 1 (interface device)			
2	One-tailed statistic	1.000	1.000	1.000
3	No. as large	109	106	94
4	One-tailed probability	0.110	0.107	0.095
5	Two-tailed statistic	1.000	1.000	1.000

6	No. as large	215	199	198
7	Two-tailed probability	0.216	0.200	0.200
	Main Effect for Factor 2 (display mode)			
8	One-tailed statistic	1.500	1.500	1.500
9	No. as large	37	38	39
10	One-tailed probability	0.038	0.039	0.040
11	Two-tailed statistic	1.500	1.500	1.500
12	No. as large	73	63	70
13	Two-tailed probability	0.074	0.064	0.071

Mean time for three runs=2 min, 57 sec

Statistical Conclusions for Design 7 (One-Tailed Main Effects) Example

Randomization tests of the main effects in a 2×2 factorial experiment on a single communication aid user were carried out. In a test of the prediction that rate of communication would be faster when the interface device was a joystick rather than a touch-screen, the proportion of 1000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.110. Therefore, the main effect of interface device was not statistically significant ($p>0.05$; one-tailed). In a test of the prediction that rate of communication would be faster when a dynamic rather than a static display mode was used, the proportion of 1000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.038. Therefore, the main effect of display mode was statistically significant ($p<0.05$; one-tailed).

Testing Simple Effects in the Factorial Design

Although there is no randomization test available for testing the interaction between interface device and display mode, if a particular form of interaction has been predicted, predictions for simple effects can be derived from the predicted interaction, and these can be tested using a randomization test. These are, of course, precisely the follow-up tests that would normally be made following the finding of a significant interaction. The tests of simple effects can be carried out using the macro for Design 5a. We present the results for a randomization test of one of the four simple effects, that predicting that a dynamic display mode will be superior to a static display mode when a touch-screen interface is used.

The data for entry into the Design 5a worksheet would be as follows. In Row 2 of Column A, the number of observations involving only the touch-screen interface (i.e., 8) is entered. Column B will contain the touch-screen data for static and dynamic display modes and the display mode codes will be entered in Column C, where the display mode predicted to result in a faster communication rate (i.e., the dynamic mode) is coded 2. The worksheet entries are shown in the Design 5a Worksheet Box (Simple Effects Example) under "Example for Design *7*" in chapter 5, and a Design 5a Workbook containing the simple effects worksheet example has been saved as *design5a_simple.xls*.

Randomization Test Results for Design 7 (Simple Effect of Display Mode With Touch-Screen Interface) Example

Row	*Result*	*1st run Col. 1*	*2nd run Col.*	*3rd run Col. 1*
2	One-tailed statistic	3.250	3.250	3.250
3	No. as large	18	9	14
4	One-tailed probability	0.019	0.010	0.015
5	Two-tailed statistic	3.250	3.250	3.250
6	No. as large	30	26	31
7	Two-tailed probability	0.031	0.027	0.032

Mean time for three runs=1 min, 45 sec

Statistical Conclusion for One-Tailed Test of a Simple Effect

Predictions for simple effects were derived from the predicted form of the interaction between interface device and display mode. In a randomization test of the prediction that a dynamic display mode would result in a faster communication rate than a static display mode when the interface device was a touch-screen, the proportion of 1000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.019. Therefore, the simple effect of dynamic display with a touch-screen interface was statistically significant ($p<0.05$; one-tailed).

DESIGN 8 (ORDINAL PREDICTIONS WITHIN NONEXPERIMENTAL DESIGNS)

The randomization test that we present for application to a range of designs involving ordinal predictions is only valid in a strict sense when a genuine random assignment procedure has been incorporated in the design. We believe that there may be some circumstances in which use of the randomization test is justified even though a genuinely random procedure for the assignment of experimental units to observation periods has not been followed. We trust that those interested in using the test will read our discussion of the issue in chapter 10 and come to a view of the appropriateness of the test for analysis of their own data.

Example data for a predicted unique order and for a predicted partial order were presented in chapter 5. The macro containing the data for the unique order prediction is on the CD-ROM in the file named *design8unique.xls*. Another macro, identical apart from containing the data for the partial order prediction, is named *design8partial.xls*.

Specifications for Design 8 Example

Number of participants (or number of observations on a single participant) = 6

Predicted order: Unique or partial

Commented Macro for Design 8 (Macro File Name: design8unique.xls or design-8partial.xls)

```
Sub design8()
Columns("D:J").Clear-Contents                    clear the columns to be used
Dim j As Integer                                 this will count arrangements
Dim lastsubject As Integer                       number of participants
lastsubject=Range("A2")
Dim lastrow$                                     this is the last row of data and
                                                 associated columns
lastrow$=lastsubject+1
Range("B2:B" & lastrow$).Select
Selection.Copy                                   make a copy of the
                                                 observations to be arranged
Range("D2").Select
ActiveSheet. Paste
For j=1 To 1000                                  number of arrangements
Range("E2").Select                               fill a column with random
                                                 numbers
ActiveCell.FormulaR1C1="=Rand()"
Selection.AutoFill Destination:=Range("E2:E" & lastrow$)
Range("D2:E" & lastrow$).Select                  sort the random numbers and
                                                 carry along the copy of the
                                                 observations to give an
                                                 arrangement
Selection.Sort Key1:=Range("E2"), Order1:=xlAscending, _
Header:=xlGuess, OrderCustom:=1, MatchCase:=False, _
Orientation:=xlTopToBottom
Range("F2").Select                               multiply the arrangement with
                                                 the predicted order
ActiveCell.FormulaR1C1="=RC[-2]*RC[-3]"
Selection.AutoFill Destination:=Range("F2:" & lastrow$)
Range("G2").Select                               sum the products for this
                                                 arrangement
ActiveCell.FormulaR1C1="=SUM(C[-1])"
Selection.Copy
Range("H2").Select                               and store it
Selection.Insert Shift:=xlDown
Selection.PasteSpecial Paste:=xlValues
Next                                             next arrangement
Range("l2").Select                               find the products of
                                                 observations and predicted
                                                 order
```

```
ActiveCell.FormulaR1C1="=RC[−6]*RC[−7]"
Selection.AutoFill Destination:=Range("l2:l" & lastrow$)
Range("J2").Select                                  sum the products for the
                                                    predicted order
ActiveCell.FormulaR1C1="=SUM(C[−1])"                count the arrangements with a test
Range("J3").Select                                  statistic at least as large
ActiveCell.FormulaR1C1="=COUNTIF(C[−2]:C[−2],""">=""&R[−1]C)"
Range("J4").Select                                  find the probability
ActiveCell.FormulaR1C1="=(R[−1]C+1)/(1000+1)" End Sub
```

Location of Design 8 Test Results

After running, the test statistic (for the actual data) is in Cell J2. In J3 is the count of arrangement test statistics at least as large and in J4 is the one-tailed probability.

Randomization Test Results for Design 8 Example

Row	*Result*	*1st run Col.*	*2nd run Col.*	*3rd run Col.*
	Test of the Unique Order Prediction			
2	One-tailed statistic	110.000	110.000	110.000
3	No. as large	22	10	16
4	One-tailed probability	0.023	0.011	0.017
	Test of the Partial Order Prediction			
2	One-tailed statistic	44.000	44.000	44.000
3	No. as large	90	97	104
4	One-tailed probability	0.091	0.098	0.010

Mean time for six runs=57 sec

Statistical Conclusion for Design 8 (Unique Order Prediction of Small Group Data) Example

In a randomization test of the prediction of a unique ordering of communicative competence ratings of communication aid users, the proportion of 1000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.023. Therefore, the correlation between the obtained order and the predicted order (based on the etiology of speech impairment and severity of additional physical impairments) was statistically significant ($p<0.05$; one-tailed).

Statistical Conclusion for Design 8 (Partial Order Prediction of Single-Case Data) Example

In a randomization test of the prediction of a partial ordering of frequency of questions asked by a communication aid user of same-sex and opposite-sex partners, the proportion of 1000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.091. Therefore, the correlation between the obtained order and the predicted order (based on the gender of partners) was not statistically significant ($p>0.05$; one-tailed).

Chapter 8
Randomization Tests Using SPSS

All of the designs described in chapter 5 can be subjected to randomization tests within SPSS for Windows using the same general procedures. We suggest that you begin by copying all of the files in the SPSS subdirectory on the companion CD-ROM to a convenient directory. There are two kinds of files there: SPSS programs (or syntax documents; for the sake of consistency across the three packages, we refer to them elsewhere throughout the book as macros) with .sps extensions (*design1.sps* to *design8.sps*) and SPSS data files with .sav extensions (*design1.sav* to *simple.sav*). The next step, once this has been done, is to edit the data worksheet that corresponds to the design of interest so that it contains your own data, remaining in conformity with the worksheet specifications given in chapter 5. To edit one of the worksheets (e.g., design1.sav to analyze Design 1 data) use *Open* from the *File* menu to open it in SPSS and simply replace the numerical entries in all columns with your own values. Then, if you wish, use *Save As* from the *File* menu to save the current worksheet in the same directory, calling it owndes1 or something similar (the .sav extension will be added automatically). To run the program, with the worksheet open, open the program using *Open* from the *File* menu. You will need to scroll down the *Types of File* list to *Syntax(*.sps)* and then select the program you want by double-clicking on the file name in the list that appears. As the extensions are not displayed, the program names and data files may appear identical in the **.sav* and **.sps* lists, so you need to make sure that you have selected from the **.sps* list. If you do not do that, you will probably get an error message telling you that the data file is already open. To run the pro-gram, select *Run* and then *All* from the *Syntax Editor Window*. After running, the results will appear in the *Output Viewer* window. These will comprise the value of the statistic used for the actual data, the count of arrangement statistics at least as large (in some cases, at least as small) as the statistic for the actual data, and the probability of obtaining a statistic as extreme as the one obtained. SPSS presents small probabilities in scientific notation.Thus, for example:

```
10**−2X
1.499250375
```

should be read as:

```
0.01499250375
```

by moving the decimal point two places to the left (indicated by the −2).

Where one-tailed and two-tailed probabilities are available, these are clearly labeled as such. It should be noted that in some of the designs, one- and two-tailed probabilities often, but not necessarily, will be identical. For example, in the single-case AB design (Design 1), the probabilities will only differ when one or more arrangements of the data produces a difference between means in the nonpredicted direction that is at least as large as the observed difference between means.

All our SPSS programs use the SPSS matrix language. They all start by setting the maximum number of loops allowed to 5000, as the default is only 40. The maximum can be increased if required. The number of data arrangements to be randomly sampled is set at 2000 in the program (2001, including the actual data). It can be changed by using *Edit* and *Find* to locate 2001 in the program and substituting the required value (e.g., 5001). These, and any other required modifications to the program, can be carried out in the Syntax Editor window. After editing, the program should be saved as an **.sps* file with the same or a different name as the user wishes.

In the remainder of the chapter, the following are listed for each design:

1. The design specifications for the example in chapter 5.
2. The program for the design, with bolded comments indicating the function of each section of the program.
3. The randomization test results for three runs of the program on the example worksheet, including the average time taken for the run using SPSS for Windows on a Pentium P266 MHZ with 64 Mb RAM and an Intel processor, with no other packages running concurrently (all of the designs took less than 1 min to run).
4. A statement of the statistical conclusion based on the first run of the program (which normally would be the only result obtained).

It should be noted that the count of rearrangement statistics that are at least as large (or in some cases, at least as small) as the test statistic, and its probability under the null hypothesis, will vary slightly from run to run on the same data. This is due to random sampling fluctuations as 2000 arrangements are randomly sampled. Provision of the results of three runs of the program should help users to gain some idea of the extent of variation using 2000 samples. We have generally provided sample data that would give probabilities close to a critical value (e.g., p=0.05) if systematic sampling of all possible arrangements were generated. For example, in Design 1, systematic generation of all 21 possible arrangements of the data would yield a probability of 1/21=0.048. Consequently, using random sampling of arrangements with replacement, the statistical decisions obtained sometimes differ between runs with the same data. If your own data produce probabilities close to a critical value, you may wish to obtain more stable probability estimates than those provided by 2000 samples. However, the propriety of sequential testing of this kind in the absence of taking any steps to protect the significance level is open to question and, ideally, a decision about how many samples to use should be made at the outset (see the discussion of this issue in chap. 12). If you nonetheless decide to pursue a sequential strategy or you decide at the outset to use a different number of samples, you should open the program in the syntax window as described earlier, locate the 2001 entry and change it to, say, 5001. You can, of course, save the edited program with a new name if you wish. Increasing the number of random samples will increase the length of the run time, so it may be a good idea to try a run, possibly using simulated data comparable to your real data, with the original 2000 rearrangements on your own computer to see how long that takes before making a change.

SPSS appears to have a problem recognizing exact equality in some unusual cases. Because we have to compare actual test statistics with those obtained by data rearrangements, we have solved this problem by subtracting from each actual test statistic a very

small multiple of itself, in fact 10^{-6} times itself. This information is for the benefit of anyone wishing to write his or her own randomization test programs using SPSS. It can be disregarded by anyone using our programs.

DESIGN 1 (AB)

Specifications for Design 1 Example

Total number of observation periods	=	36
Minimum number of baseline periods	=	8
Minimum number of treatment periods	=	8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention point for the example is at Period 15.

Commented Program for Design 1 (Program File Name: design1.sps)

```
set mxloop 5000.                          increase the maximum loop size to 5000
compute phase=phase+1.                    change the phase labeling to 1 and 2 so that they can
                                          be column numbers in the data matrix
matrix.                                   start the matrix language form a matrix from the
get limits/variables=limits/              first column of the data window
missing=omit.
get data/variables=data phase/missing=omit.
                                          form a matrix from the second and third columns of
                                          the data window
compute ncase=limits(1).                  collect the number of rows of data
compute totals={0;0}.                     start baseline and intervention totals and counts at
                                          zero
compute counts={0;0}.                     find the totals and counts for baseline and interven-
loop case=1 to ncase.                     tion for the actual data
compute totals(data(case, 2))=totals(data(case, 2))+data(case, 1).
compute counts(data(case, 2))=counts(data(case, 2))+1.
end loop.
compute means=totals/counts.              and the means and the test statistic
compute test=means(2, 1)−means(1,
1).
print test/title="test statistic".        this is the number of arrangements +1 for the actual
compute nperm=2001.                       data
compute results=uniform(nperm, 1).        make a matrix of the correct shape to receive the
                                          results—it is full of random numbers but
                                                  will be overwritten later
```

```
compute results(1, 1)=test-test/1000000.
compute pos1=0.
compute pos2=0.
```

put the actual test statistic in the first place in the results matrix, reduced by a very small multiple of itself to avoid comparison problems
this will be the count of arrangement statistics at least as large as the actual test statistic and this will be the count of arrangement statistics at least as large in absolute value as the absolute value of the actual test statistic

```
loop perm=2 to nperm.
```

now start the rearrangements (the first is just the actual data)

```
compute rand=uniform(1, 1).
```

a random number to choose the intervention point

```
compute temp1=limits(1)−limits(2)−limits(3)+1.
compute temp2=limits(2)+1.
```

we need to make the intervention fall in the permitted range

```
compute inter=trunc(temp1*rand)+temp2.
compute totals={0;0}.
```

start baseline and intervention totals and counts at zero

```
compute counts={0;0}.
loop case =1 to inter−1.
compute totals(1)=totals(1)+data(case,1).
compute counts(1)=counts(1)+1.
end loop.
```

collect total and count for baseline

```
loop case=inter to ncase.
compute totals(2)=totals(2)+data(case,1).
compute counts(2)=counts(2)+1.
end loop.
```

and intervention

```
compute means=totals/counts.
compute test=means(2, 1)−means(1,1).
compute results(perm,1)=test.
end loop.
```

find the means and test statistic

and put in the results matrix
next arrangement

```
compute absres=abs(results).
loop k=2 to nperm.
```

find absolute values for two-tailed test

now compare arrangement test statistics with the actual one, and

count those at least as large

```
do if results(k, 1)>=results(1, 1).
compute pos1=pos1+1.
end if.
```

```
do if absres(k, 1)>=absres(1, 1).                    and for absolute values
compute pos2=pos2+1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large".
compute prob1=(pos1+1)/nperm.                        calculate the one-tailed
print prob1/title="one tail probability".            probability

print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)".
compute prob2=(pos2+1)/nperm.                        and the two-tailed probability
print prob2/title="two tail probability".
end matrix.                                          end of the matrix language
compute phase=phase−1.                               restore the phase labels
execute.
```

Randomization Test Results for Design 1 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	2.656	2.656	2.656
No. as large	98	100	89
One-tailed probability	0.049	0.050	0.045
No. as large in absolute value	98	100	89
Two-tailed probability	0.049	0.050	0.045

Mean time for three runs=10 sec

Statistical Conclusion for Design 1 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would increase when a word prediction system was introduced, the proportion of 2000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.049. Therefore, the obtained difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 2 (ABA REVERSAL)

Specifications for Design 2 Example

Total number of observation periods	=	36
Minimum number of control periods before treatment starts	=	8
Minimum number of treatment periods	=	8
Minimum number of control periods after withdrawal of treatment	=	8

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pair of intervention and withdrawal points for the example is at Periods 12 and 25.

Commented Program for Design 2 (Program File Name: design2.sps)

```
set mxloop 5000.                          increase the maximum loop size to 5000
compute phase=phase+1.                    change the phase labeling to 1 and 2 so that they
                                          can be column numbers in the data matrix
matrix. get limits/variable=limits/       start the matrix language form a matrix from the
missing=omit.                             first column of the data window
get data/variables=data phase/
missing=omit.
                                          form a matrix from the second and third columns
                                          of the data window
compute ncase=limits(1).                  collect the number of rows of data
compute totals={0;0}.                     start baseline/withdrawal and intervention totals
                                          and counts at zero
compute counts={0;0}. loop case=1 to      find the totals and counts for baseline/withdrawal
ncase.                                    and intervention for the actual data
compute totals(data(case, 2))=totals(data(case, 2))+data(case, 1).
compute counts(data(case, 2))=counts(data(case, 2))+1.
end loop.
compute means=totals/counts.              and the means
compute test=means(2, 1)−means(1, 1).     and the test statistic
print test/tltle="test statistic".
compute nperm=2001.                       this is the number of
                                          arrangements+for the actual
                                          data
compute results=uniform(nperm, 1).        make a matrix of the correct
                                          shape to receive the results—it
                                          is full of random numbers but
                                          will be overwritten later
compute results(1, 1)=test-test/1000000.  put the actual test statistic in the
                                          first place in the results matrix,
                                          reduced by a very small multiple
                                          of itself to avoid comparison
                                          problems
compute pos1=0.                           this will be the count of
                                          arrangement statistics at least as
                                          large as the actual test statistic
```

```
compute pos2=0.                                  and this will be the count of
                                                 arrangement statistics at least as
                                                 large in absolute value as
                                                 the absolute value of the actual
                                                 test statistic
loop perm=2 to nperm.                            now start the rearrangements
                                                 (the first is just the actual data)
compute rand=uniform(2, 1).                      a pair of random numbers to
                                                 choose the intervention and
                                                 withdrawal points
compute temp1=limits(2)+1.                       we need to make the intervention
                                                 fall in the permitted range
compute temp2=limits(1)−limits(2)−limits(3)−limits(4)+1.
compute interven=trunc(temp2*rand(1))+temp1.
compute temp3=limits(1)−limits(4).               and we need to make the
                                                 withdrawal fall in the permitted
                                                 range given the intervention
compute temp4=interven+limits(3).
compute temp5=temp3−temp4+2.
compute withdraw=trunc(temp5*rand(2))+interven+limits(3).
compute totals={0;0}.                            start baseline/withdrawal and
                                                 intervention totals and counts
                                                 at zero
compute counts={0;0}.
loop case=1 to interven-1.                       collect total and count for
                                                 baseline
compute totals(1)=totals(1)+data(case,1).
compute counts(1)=counts(1)+1.                   collect total and count for
end loop.                                        intervention
loop case=interven to withdraw−1.
compute totals(2)=totals(2)+data(case,1).
compute counts(2)=counts(2)+1.                   collect total and count for
end loop.                                        withdrawal
loop case=withdraw to ncase.
compute totals(1)=totals(1)−data(case,1).
compute counts(1)=counts(1)+1.
end loop.
```

```
compute means=totals/counts.                    find the means and test statistic
compute test=means(2,1)−means(1,1).
compute results(perm,1)=test.
end loop.                                       and put in the results matrix
compute absres=abs(results).                    next arrangement
loop k=2 to nperm.                              find absolute values for
do if results(k,1)>=results(1,1).               two-tailed test
compute pos1=pos1+1.                            now compare arrangement test
end if.                                         statistics with the actual one, and
do if absres(k,1)>=absres(1,1).                 count those at least as large
compute pos2=pos2+1.
end if.
end loop.                                       and for absolute values

print pos1/title="count of arrangement statistics at least as large".
compute prob1=pos1+1)/nperm.                    calculate the one-tailed
                                                probability
print prob1/title="one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)"
compute prob2=(pos2+1)/nperm.                   and the two-tailed probability
print prob2/title="two tail probability".
end matrix.                                     end of the matrix language
compute phase=phase-1.                          restore the phase labels
execute.
```

Randomization Test Results for Design 2 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	2.592	2.592	2.592
No. as large	81	74	82
One-tailed probability	0.041	0.037	0.041
No. as large in absolute value	81	74	82
Two-tailed probability	0.041	0.037	0.041

Mean time for three runs=12 sec

Statistical Conclusion for Design 2 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that a communication aid user's rate of text entry would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 2000 randomly sampled data divisions giving a rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.041. Therefore, the obtained difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 3 (AB MULTIPLE BASELINE)

Specifications for Design 3 Example

Number of participant (or behavior) replications (i.e., baselines)	=	3
Total number of observation periods per participant (or behavior)	=	18
Minimum number of control periods before treatment starts	=	6
Minimum number of treatment periods	=	6

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected intervention points for the example are 9,12, and 8.

Commented Program for Design 3 (Program File Name: design3.sps)

```
set mxloop 5000.                                  increase the maximum loop size
                                                  to 5000
compute phase=phase+1.                            change the phase labeling to 1
                                                  and 2 so that they can be column
                                                  numbers in the data matrix
matrix.                                           start the matrix language
get limits/varJable=limits/missing=omit.          form a matrix from the first
                                                  column of the data window
get data/variables=data phase subject/missing=omit.
                                                  form a matrix from the second,
                                                  third and fourth columns of the
                                                  data window
compute nsubject=limits(1).                       collect the number of participants
compute ncase=limits(2)*limits(1).                compute the number of rows of
                                                  data
compute totals=make(nsubject,2,0).                make a matrix of zeros of the
                                                  correct shape to
compute counts=make(nsubject,2,0).                receive the baseline and
                                                  intervention totals
loop case=1 to ncase.                             and counts
                                                  find the totals and counts for
                                                  baseline and intervention for the
                                                  actual data
compute totals(data(case,3),data(case,2))=toals(data(case,3),data(case,2))+data(case,1).
compute counts(data(case,3),data(case,2))=counts(data(case,3),data(case,2))+1.
end loop.
compute means=totals/counts.                      and the means
compute test=0.                                   and the test statistic, totaling
                                                  over participants
```

```
loop k=1 to nsubject.
compute test=test+means(k,2)−means(k,1).
end loop.
print test/title="test statistic",
compute nperm=2001.                          this is the number of
                                             arrangements +1 for the actual
                                             data
compute results=uniform(nperm,1).            make a matrix of the correct
compute results(1,1)=test-test/1000000.      shape to receive the results—it
                                             is full of random numbers but
                                             will be overwritten later
                                             put the actual test statistic in the
                                             first place in the results matrix,
                                             reduced by a very small multiple
                                             of itself to avoid comparison
                                             problems
compute pos1=0.                              this will be the count of
compute pos2=0.                              arrangement statistics
                                             at least as large as the actual test
                                             statistic
                                             and this will be the count of
                                             arrangement statistics at least as
                                             large in absolute value as the
                                             absolute value of the actual test
                                             statistic
loop perm=2 to nperm.                        now start the rearrangements
compute totals=0*totals.                     (the first is just the actual data)
                                             start baseline and intervention
                                             totals and counts at zero, and the
                                             test statistic, for this
                                             arrangement
compute counts=0*counts.
compute test=0.
compute rand=uniform(nsubject,1).            a set of random numbers to
loop k=1 to nsubject.                        choose the intervention points
                                             for each participant
                                             do the intervention and test
                                             statistic for each
                                             participant in this data
                                             arrangement
compute temp1=limits(2)−limits(3)−limits(4)+1.
compute interven=trunc(temp1*rand(k))+limits(3)+1.
                                             we need to make the
                                             intervention fall in the permitted
                                             range
```

```
compute temp2=(k−1)*limits(2) +1.
compute temp3=temp2+interven−1.
compute temp4=k*limits(2).                    collect total and count for
loop case=temp2 to temp3.                     baseline for each participant
compute totals(k,1)=totals(k,1)+data(case,1).
compute counts(k,1)=counts(k,1)+1.
end loop,
loop case=temp3+1 to temp4.                   collect total and count for
                                              withdrawal for each participant
compute totals(k,2)=totals(k,2)+data(case,1).
compute counts(k,2)=counts(k,2)+1.
end loop,
compute test=test+totals(k,2)/counts(k,2)−totals(k,1)/counts(k,1).
                                              add on the test statistic for the
                                              current participant
end loop.                                     next participant
compute results(perm,1)=test.                 save the test statistic for this
                                              arrangement
end loop.                                     next arrangement
compute absres=abs(results).                  find absolute values for
                                              two-tailed test
loop k=2 to nperm.                            now compare arrangement test
                                              statistics with the actual one, and
                                              count those at least as large
do if results(k,1)>=results(1,1).
compute pos1 =pos1+1.                         and for absolute values
end if.
do if absres(k,1)>=absres(1,1).
compute pos2=pos2+1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large",
compute prob1=(pos1+1)/nperm.                 calculate the one-tailed
                                              probability
print prob1/title="one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)"
compute prob2=(pos2+1)/nperm.                 and the two-tailed probability
print prob2/title="two tail probability".     end of the matrix language
end matrix.                                   restore the phase labels
compute phase=phase-1.
execute.
```

Randomization Test Results for Design 3 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	5.915	5.915	5.915
No. as large	59	56	54
One-tailed probability	0.030	0.028	0.027
No. as large in absolute value	59	56	54
Two-tailed probability	0.030	0.028	0.027

Mean time for three runs=20 sec

Statistical Conclusion for Design 3 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of three communication aid users would increase when a word prediction system was introduced, the proportion of 2000 randomly sampled data divisions giving a combined rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.030. Therefore, the obtained summed difference in text entry rate before and after introduction of a word prediction system was statistically significant ($p<0.05$; one-tailed).

DESIGN 4 (ABA MULTIPLE BASELINE)

Specifications for Design 4 Example

Number of participants	=	2
Total number of observation periods per participant	=	12
Minimum number of control periods before treatment starts	=	3
Minimum number of treatment periods	=	3
Minimum number of control periods after withdrawal of treatment	=	3

The one-tailed test assumes the intervention increases scores (see chap. 5, "Example for Design 1," if this assumption is not true for your data).

The randomly selected pairs of intervention and withdrawal points for the example are as follows: Participant 1 at Periods 6 and 10; Participant 2 at Periods 4 and 8.

Commented Program for Design 4 (Program File Name: design4.sps)

```
set mxloop 5000.                    increase the maximum loop size
                                    to 5000
compute phase=phase+1.              change the phase labeling to 1
matrix.                             and 2 so that they can be column
                                    numbers in the data matrix
                                    start the matrix language
```

```
get limits/variable=limits/missing=omit.
```
form a matrix from the first column of the data window

```
get data/variables=data phase subject/missing=omit.
```
form a matrix from the second, third and fourth columns of the data window

```
compute nsubject=limits(1).
```
collect the number of participants

```
compute ncase=limits(2)*limits(1).
```
compute the number of rows of data

```
compute totals=make(nsubject,2,0).
```
make a matrix of zeros of the correct shape to receive the baseline/withdrawal and intervention totals

```
compute counts=make(nsubject, 2, 0).
```
and counts

```
loop case =1 to ncase.
```
find the totals and counts for baseline and intervention for the actual data

```
compute totals(data(case,3),data(case,2))=totals(data(case,3),data(case,2))+data(case,1).
compute counts(data(case,3),data(case,2))=counts(data(case,3),data(case,2))+1.
end loop.
```

```
compute means=totals/counts.
```
and the means

```
compute test=0.
```
and the test statistic, totaling over participants

```
loop k=1 to nsubject.
compute test=test+means(k,2)−means(k,1).
end loop.
print test/title="test statistic",
```

```
compute nperm=2001.
```
this is the number of arrangements+1 for the actual data

```
compute results=uniform(nperm,1).
```
make a matrix of the correct shape to receive the results—it is full of random numbers but will be overwritten later

```
compute results(1,1)=test-test/1000000.
```
put the actual test statistic in the first place in the results matrix, reduced by a very small multiple of itself to avoid comparison problems

```
compute pos1=0.
```
this will be the count of arrangement statistics at least as large as the actual test statistic

```
compute pos2=0.
```
and this will be the count of arrangement statistics at least as large in absolute value as the absolute value of the actual test statistic

```
compute temp1=limits(3)+1.
compute temp2=limits(2)−limits(3)−limits(4)−limits(5)+1.
compute temp3=limits(2)−limits(5).
loop perm=2 to nperm.                          now start the rearrangements
                                               (the first is just the actual data)
compute interven=uniform(nsubject,2).          a matrix of random numbers for
                                               the intervention and withdrawal
                                               for each participant
compute totals=0*totals.                       start baseline/withdrawal and
                                               intervention totals and counts at
                                               zero, and the test statistic, for
                                               this arrangement
compute counts=0*counts.
compute test=0.
loop k=1 to nsubject.                          do the intervention and
                                               withdrawal, and test statistic for
                                               each participant in this data
                                               arrangement
compute interven(k,1)=trunc(temp2*interven(k,1))+temp1.
                                               we need to make the intervention
                                               fall in the permitted range
compute temp4=interven(k,1)+limits(4).
compute temp5=temp3−temp4+2.
compute Jnterven(k,2)=trunc(temp5*interven(k,2))+temp4.
                                               we need to make the withdrawal
                                               fall in the permitted range given
                                               the intervention
compute temp6=(k−1)*limits (2)+1.
compute temp7=temp6+interven(k,1)−2.              collect total and count for
compute temp8=temp6+interven(k,2)−2.              baseline for each participant
compute temp9=k*limits(2).
loop case=temp6 to temp7.
compute totals(k,1)=totals(k,1)+data(case,1).
compute counts(k,1)=counts(k,1)+1.
end loop,
loop case=temp7+1 to temp8.                    collect total and count for
compute totals(k,2)=totals(k,2)+data(case,1).  intervention for each participant
compute counts(k,2)=counts(k,2)+1.
end loop,
loop case=temp8+1 to temp9.                    collect total and count for
compute totals(k,1)=totals(k,1)+data(case,1).  withdrawal for each participant
compute counts(k, 1)=counts(k, 1)+1.
end loop,
compute test=test+totals(k,2)/counts(k,2)−totals(k,1)/counts(k,1).
```

```
                                        add on the test statistic for the
                                        current participant
end loop.                               next participant
compute results(perm,1)=test.           save the test statistic for this
end loop.                               arrangement
compute absres=abs(results).            next arrangement
loop k=2 to nperm.                      find absolute values for
                                        two-tailed test
                                        now compare arrangement test
                                        statistics with the actual one, and
                                        count those at least as large
do if results(k,1)>=results(1,1).
compute pos1=pos1+1.
end if.
do if absres(k,1)>=absres(1,1).         and for absolute values
compute pos2=pos2–1 1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large",
compute prob1=(pos1+1)/nperm.           calculate the one-tailed
                                        probability
print prob1/title="one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)".
compute prob2=(pos2+1)/nperm.           and the two-tailed probability
print prob2/title="two tail probability".
end matrix.                             end of the matrix language
compute phase=phase-1.                  restore the phase labels
execute.
```

Randomization Test Results for Design 4 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	3.750	3.750	3.750
No. as large	58	53	44
One-tailed probability	0.029	0.027	0.022
No. as large in absolute value	58	53	44
Two-tailed probability	0.029	0.027	0.022

Mean time for three runs=13 sec

Statistical Conclusion for Design 4 Example (Assuming a Directional Prediction)

In a randomization test of the prediction that the summed rates of text entry of two communication aid users would be faster when a word prediction system was used than in control phases before its introduction and after its withdrawal, the proportion of 2000 randomly

sampled data divisions giving a combined rate difference in the predicted direction at least as large as the experimentally obtained difference was 0.029. Therefore, the obtained summed difference in text entry rate using a word prediction system compared with the rate before and after its introduction was statistically significant ($p<0.05$; one-tailed).

DESIGN 5 (ONE-WAY SMALL GROUPS AND SINGLE-CASE RANDOMIZED TREATMENT)

Specifications for Design 5 Example

Total number of participants (or treatment occasions)	=	10
Number of treatments: Communication aid systems (or levels of symbol translucency)	=	4
Number of participants per communication system (or symbol sets per translucency level)	=	2, 2, 3, 3

Commented Program for Design 5 (Program File Name: design5.sps)

```
set mxloop=5000.                                  increase the maximum loop size to 5000
matrix.                                           start the matrix language
get limits/variables=limits/missing=omit.         form a matrix from the first column of the data window
get data/variables=data condit/missing=omit.      form a matrix from the second and third columns of the data window
compute ncase=limits(1).                          collect the number of rows of data
compute colmax=cmax(data).
compute ngroup=colmax(2).                         find the number of treatment groups
compute totals=make(ngroup,1,0).                  make a matrix of zeros of the correct shape to receive the treatment totals
compute counts=make(ngroup,1,0).                  and counts
loop case=1 to ncase.                             find the totals and counts for treatments for the actual data
compute totals(data(case,2))=totals(data(case,2))+data(case,1).
compute counts(data(case,2))=counts(data(case,2))+1.
end loop.
compute TrSS=0.                                   collect the treatment SS for the actual data
```

```
loop k=1 to ngroup.
compute TrSS=TrSS+totals(k)*totals(k)/counts(k).
end loop.
compute TSS=cssq(data).                          find the total SS
compute RSS=TSS(1)−TrSS.                         and the RSS
print RSS/title="RSS".
compute nperm=2001.
                                                 this is the number of
                                                 arrangements +1 for the actual
                                                 data
compute results=uniform(nperm,1).                make a matrix of the correct
                                                 shape to receive the results—it
                                                 is full of random numbers but
                                                 will be overwritten later
compute results(1,1)=RSS+RSS/1000000.            put the actual test statistic in the
                                                 first place in the results matrix,
                                                 increased by a very small
compute pos=0.                                   multiple of itself to avoid
                                                 comparison problems
loop perm=2 to nperm.                            this will be the count of
                                                 arrangement statistics at least as
                                                 small as the actual test statistic
loop case=1 to ncase.                            now start the rearrangements
                                                 (the first is just the actual data)
                                                 this loop shuffles the data
compute k=trunc(uniform(1,1)*(ncase−case+1))+case.
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
compute totals=0*totals.                         start treatment totals and counts
                                                 at zero
compute counts=0*counts.
loop case=1 to ncase.                            collect totals and counts for
                                                 treatments for this arrangement
compute totals(data(case,2))=totals(data(case,2))+data(case,1).
compute counts(data(case,2))=counts(data(case,2))+1. end loop.
compute TrSS=0.                                  collect the treatment SS for this
                                                 arrangement
loop k=1 to ngroup.
compute TrSS=TrSS+totals(k)*totals(k)/counts(k).
end loop.
compute TSS=cssq(data).                          find the total SS for this
                                                 arrangement
```

```
compute RSS=TSS(1)-TrSS.                    and the RSS for this arrangement
compute results(perm,1)=RSS.                and put in the results matrix
end loop.                                   next arrangement
loop k=2 to nperm.                          now compare arrangement test statistics with the actual one, and count those at least as small
do if results(k,1)<=results(1,1).
compute pos=pos+1.
end if.
end loop.
print pos/title="count of arrangement RSS at least as small".
compute prob=(pos+1)/nperm.                 calculate the probability
print prob/title="probability".
end matrix.                                 end of the matrix language
```

Randomization Test Results for Design 5 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	3.333	3.333	3.333
No. as small	6	12	13
Two-tailed probability	0.003	0.006	0.007

Mean time for three runs=13 sec

Note that the test statistic is RSS and this means that the obtained value must be among the smaller values for statistical significance.

Statistical Conclusion for Design 5 (One-Way Small Groups) Example

In a randomization test of the prediction that four communication aids would differ in the time taken to learn to use them to a criterion, the proportion of 2000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.003. Therefore, the obtained differences in learning time with the four communication aids was statistically significant ($p<0.01$).

DESIGN 5a (SMALL GROUPS OR SINGLE-CASE—TWO RANDOMIZED TREATMENTS)

Specifications for Design 5a Example

Total number of participants (or treatment occasions)	=	9
Number of treatments: Communication aid systems (or levels of symbol translucency)	=	2

Number of participants per communication system (or symbol sets per translucency level) = 4, 5

For a one-tailed test the level with the higher expected mean is coded 2 and the level with the lower expected mean is coded 1.

Commented Program for Design 5a (Program File Name: design5a.sps)

```
set mxloop 5000.                                   increase the maximum loop size to 5000
matrix.                                            start the matrix language
get limits/variables=limits/missing=omit.          form a matrix from the first column of the data window
get data/variables=data condit/missing=omit.       form a matrix from the second and third columns of the data window
compute ncase=limits(1).                           collect number of rows of data
compute totals={0;0}.                              start the two treatment totals and counts at zero
compute counts={0;0}.                              find the totals and counts for treatments for the actual data
loop case=1 to ncase.
compute totals(data(case,2))=totals(data(case,2))+data(case,1).
compute counts(data(case,2))=counts(data(case,2))+1.
end loop.
compute means=totals/counts.                       and the means
compute test=means(2,1)−means(1,1).                and the test statistic
print test/title="test statistic".
compute nperm=2001.                                this is the number of arrangements +1 for the actual data
compute results=uniform(nperm,1).                  make a matrix of the correct shape to receive the results—it is full of random numbers but will be overwritten later
compute results(1,1)= test−test/1000000.           put the actual test statistic in the first place in the results matrix, reduced by a very small multiple of itself to avoid comparison problems
```

```
compute pos1=0.                                  this will be the count of
                                                 arrangement statistics at least as
compute pos2=0.                                  large as the actual test statistic
loop perm=2 to nperm.                            this will be the count of
loop case=1 to ncase.                            arrangement statistics
                                                 at least as large in absolute value
                                                 as the absolute value of the
                                                 actual test statistic
                                                 now start the rearrangements
                                                 (the first is just the actual data)
                                                 this loop shuffles the data
compute k=trunc(uniform(1,1)*(ncase−case+1))+case.
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
compute totals={0;0}.                            start treatment totals and counts
                                                 at zero
compute counts={0;0}.
loop case=1 to ncase.                            collect totals and counts for
                                                 treatments for this arrangement
compute totals(data(case,2))=totals(data(case,2))+data(case,1).
compute counts(data(case,2))=counts(data(case,2))+1.
end loop.
compute means=totals/counts.                     and the means
compute test=means(2,1)−means(1,1).              and the test statistic
compute results (perm,1)=test.                   and put in the results matrix
end loop.                                        next arrangement
compute absres=abs(results).                     find absolute values for
                                                 two-tailed test
loop k=2 to nperm.                               now compare arrangement test
                                                 statistics with the actual one, and
                                                 count those at least as large
do if results(k,1)>=results(1,1).
compute pos1 =pos1 +1.
end if.
do if absres(k,1)>=absres(1,1).                  and for absolute values
compute pos2=pos2+1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large".
compute prob1=(pos1+1)/nperm.                    calculate the one-tailed
                                                 probability
print prob1/tltle="one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)"
```

```
compute prob2=(pos2+1)/nperm.
print prob2/title="two tail probability".
end matrix.
```

and the two-tailed probability

end of the matrix language

Randomization Test Results for Design 5a Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	1.550	1.550	1.550
No. as large	74	90	69
One-tailed probability	0.037	0.045	0.035
No. as large in absolute value	143	156	142
Two-tailed probability	0.072	0.078	0.071

Mean time for three runs=12 sec

Statistical Conclusion for Design 5a (One-Tailed Single-Case) Example

In a randomization test of the prediction that high-translucency symbols will take fewer sessions than low-translucency symbols for a communication aid user to learn, the proportion of 2000 randomly sampled data divisions giving a learning sessions difference in the predicted direction at least as large as the experimentally obtained difference was 0.037. Therefore, the obtained difference in learning sessions required was statistically significant ($p<0.05$; one-tailed).

It may be noted that, had a directional prediction not been made in this instance, the two-tailed test would not have shown a significant difference ($p=0.072$).

DESIGN 6 (ONE-WAY SMALL GROUP REPEATED MEASURES AND SINGLE-CASE RANDOMIZED BLOCKS)

Specifications for Design 6 Example

Number of participants (or number of blocks)	=	3
Number of conditions	=	4
Total number of observations (conditions×participants or blocks)	=	12

Each participant must receive the same number of measures (or each treatment must appear once in each block).

Commented Program for Design 6 (Program File Name: design6.sps)

```
set mxloops 5000.                                  increase the maximum loop size to 5000
matrix.                                            start the matrix language
get limits/variables=limits/missing=omit.          form a matrix from the first column of the data window
get data/variables data condit block/missing omit.
                                                   form a matrix from the second, third and fourth columns of the data window
compute ncase=limits(1).                           collect the number of rows of data
compute colmax=cmax(data).
compute nblock=colmax(3).                          find the number of blocks (or participants)
compute ntreat=colmax(2).                          find the number of treatment conditions
compute totalt=make(ntreat,1,0).                   make a matrix of zeros of the correct shape to receive the treatment totals
compute totalb=make(nblock,1,0).                   make a matrix of zeros of the correct shape to receive the block (or participant) totals
loop case=1 to ncase.                              find the totals for treatments and blocks for the actual data
compute totalt(data(case,2)) =totalt(data(case,2))+data(case,1).
compute totalb(data(case,3))=totalb(data(case,3))+data(case,1).
end loop.                                          collect the treatment SS for the actual data
computeTrSS=0.
loop tr=1 to ntreat.
compute TrSS=TrSS+totalt(tr)*totalt(tr)/nblock.
end loop.
compute BSS=0.                                     collect the block SS for the actual data
loop bl=1 to nblock.
compute BSS=BSS+totalb(bl)*totalb(bl)/ntreat.
end loop.
compute sum=csum(data).
                                                   collect the grand total for the actual data
```

```
compute corr=sum(1)*sum(1)/ncase.          and the correction term
compute tss=cssq(data).                    and the total SS
compute RSS=tss(1)−TrSS−BSS+corr.          and the RSS
print RSS/title="RSS".
compute nperm=2001.                        this is the number of
                                           arrangements+1 for the actual
                                           data
compute results=uniform(nperm,1).          make a matrix of the correct
compute results(1,1)=RSS+RSS/1000000.      shape to receive the results—it
                                           is full of random numbers but
                                           will be overwritten later
                                           put the actual test statistic in the
                                           first place in the results matrix,
                                           increased by a very small
                                           multiple of itself to avoid
                                           comparison problems
compute pos=0.                             this will be the count of
                                           arrangement statistics
                                           at least as small as the actual
                                           test statistic
loop perm=2 to nperm.                      now start the rearrangements
                                           (the first is just the actual data)
                                           these loops shuffle the data
                                           within blocks
loop bl=1 to nblock.
loop tr=1 to ntreat.
compute k=trunc(uniform(1,1)*(ntreat−tr+1))+tr+ntreat*(bl−1).
compute case=tr+ntreat*(bl−1).
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
end loop.
compute totalt=0*totalt.                   start treatment and block totals
compute totalb=0*totalb.                   at zero
loop case=1 to ncase.                      collect treatment and block
                                           totals for this arrangement
compute totalt(data(case,2))=totalt(data(case,2))+data(case,1).
compute totalb(data(case,3))=totalb(data(case,3))+data(case,1).
end loop.
computeTrSS=0.                             collect the treatment SS for this
                                           arrangement
loop tr=1 to ntreat.
compute TrSS=TrSS+totalt(tr)*totalt(tr)/nblock. end loop.
compute BSS=0.                             collect the block SS for this
                                           arrangement
```

```
loop bl=1 to nblock.
compute BSS=BSS+totalb(bl)*totalb(bl)/ntreat.
end loop.
compute sum=csum(data).                    collect the grand total for this
                                           arrangement
compute corr=sum(1)*sum(1)/ncase.          and the correction term
compute tss=cssq(data).                    find the total SS for this
                                           arrangement
compute RSS=tss(1)−TrSS−BSS+corr.          and the RSS for this
                                           arrangement
compute results(perm,1)=RSS.               and put in the results matrix
end loop.                                  next arrangement
loop k=2 to nperm.                         now compare arrangement test
                                           statistics with the actual one, and
                                           count those at least as small
do if results(k,1)<=results(1,1).          calculate the probability
compute pos=pos+1.                         end of the matrix language
end if.
end loop.
print pos/title="count of RSS as least as small".
compute prob=(pos+1)/nperm.
print prob/title="probability".
end matrix.
```

Randomization Test Results for Design 6 Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	98.500	98.500	98.500
No. as small	2	4	5
Two-tailed probability	0.001	0.002	0.003

Mean time for three runs=24 sec

Note that the test statistic is RSS and this means that the obtained value must be among the smaller values for statistical significance.

Statistical Conclusion for Design 6 (One-Way Small Groups) Example

In a randomization test of the prediction that the number of initiations made by communication aid users will differ depending on the level of experience of conversational partners, the proportion of 2000 randomly sampled data divisions giving a test statistic (RSS) at least as small as the experimentally obtained statistic was 0.001. Therefore, the obtained differences in number of user initiations with partners with different levels of experience was statistically significant ($p<0.01$).

DESIGN 6a (TWO REPEATED MEASURES ON SMALL GROUP OR SINGLE-CASE BLOCKS)

Specifications for Design 6a Example

Number of participants (or number of blocks)	=	7
Number of conditions	=	2

Each participant must receive the same number of measures (or each treatment must appear once in each block).

For a one-tailed test the condition with the higher expected mean is coded 2 and the condition with the lower expected mean is coded 1.

Commented Program for Design 6a (Program File Name: design6a.sps)

set mxloops 5000.	**increase the maximum loop size to 5000**
matrix.	**start the matrix language form a matrix from the first column of the data window**
get limJts/variables=limits/missing=omit.	
get data/variables data condit block/missing omit.	**form a matrix from the second, third and fourth columns of the data window**
compute ncase=limits(1)*2.	**find the number of rows of data find the number of blocks (or participants)**
compute nblock=limits(1).	
compute colmax=cmax(data).	
compute ntreat=colmax(2).	**find the number of treatment conditions**
compute totalt={0,0}.	**start the two treatment totals at zero**
compute totals=uniform(nblock,1).	**make a matrix of the correct shape to receive the block totals—it is full of random numbers but will be overwritten later**
compute totals=0*totals. loop case=1 to ncase.	**start the block totals at zero find the treatment and block totals for the actual data**
compute totalt(data(case,2))=totalt(data(case,2))+data(case,1).	
compute totals(data(case,3))=totals(data(case,3))+data(case,1).	
end loop.	
compute test=(totalt(2)−totalt(1))/nblock. print test/title="test statistic",	**and the test statistic**
compute nperm=2001.	**this is the number of arrangements +1 for the actual data**
compute results=uniform(nperm,1).	**make a matrix of the correct shape to receive the results—it is full of random numbers but will be overwritten later**
compute results(1,1)=test−test/1000000.	**put the actual test statistic in the first place in the results matrix, reduced by a very small**

```
                                                    multiple of itself to avoid
                                                    comparison problems
compute pos1=0.                                     this will be the count of
                                                    arrangement statistics
                                                    at least as large as the actual test
                                                    statistic
compute pos2=0.                                     this will be the count of
                                                    arrangement statistics
                                                    at least as large in absolute value
                                                    as the absolute value of the actual
                                                    test statistic
loop perm=2 to nperm.                               now start the rearrangements
loop bl=1 to nblock.                                (the first is just the actual data)
                                                    these loops shuffle the data
                                                    within blocks

loop tr=1 to 2.
compute k=trunc(uniform(1,1)*(2−tr+1))+tr+2*(bl−1).
compute case=tr+2*(bl−1).
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop,
end loop,
compute totalt={0,0}.                               start treatment and block totals
                                                    at zero

compute totals=0*totals.
loop case=1 to ncase.                               collect treatment and block
                                                    totals for this arrangement

compute totalt(data(case,2))=totalt(data(case,2))+data(case,1).
compute totals(data(case,3))=totals(data(case,3))+data(case,1).
end loop.
compute test=(totalt(2)−totalt(1))/nblock.          and the test statistic
compute results(perm,1)=test.                       and put in the results matrix
end loop.                                           next arrangement
compute absres=abs(results).                        find absolute values for
                                                    two-tailed test
loop k=2 to nperm.                                  now compare arrangement test
                                                    statistics with the actual one, and
                                                    count those at least as large

do if results(k,1)>=results(1,1).
compute pos1=pos1+1.
end If.
```

```
do if absres(k,1)>=absres(1,1).                    and for absolute values
compute pos2=pos2+1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large".
compute prob1=(pos1+1)/nperm.                      calculate the one-tailed
                                                   probability
print prob1/title="one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(test)"
compute prob2=(pos2+1)/nperm.                      and the two-tailed probability
print prob2/title="two tail probability",
end matrix.
                                                   end of the matrix language
```

Randomization Test Results for Design 6a Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	1.857	1.857	1.857
No. as large	80	112	102
One-tailed probability	0.040	0.056	0.051
No. as large in absolute value	166	181	187
Two-tailed probability	0.083	0.091	0.094

Mean time for three runs=27 sec

Statistical Conclusion for Design 6a (One-Tailed Single-Case) Example

In a randomization test of the prediction that a communication aid user will choose to use a high-tech aid more frequently with a speech and language therapist than with a family member, the proportion of 2000 randomly sampled data divisions giving a difference in high-tech aid use in the predicted direction at least as large as the experimentally obtained difference was 0.040. Therefore, the obtained difference in high-tech use was statistically significant ($p<0.05$; one-tailed).

As in Design 5a, if a directional prediction had not been made in this instance, the two-tailed test would not have shown a significant difference ($p=0.083$).

DESIGN 7 (TWO-WAY FACTORIAL SINGLE CASE)

Specifications for Design 7 Example

Factor 1=interface device (touch-screen coded 1, joystick coded 2)

Factor 2=display mode (static coded 1, dynamic coded 2)

Total number of observations	=	16
Number of observations per condition (must be equal)	=	4

Directional prediction for Factor 1: joystick > touch-screen

Directional prediction for Factor 2: dynamic > static

Predicted interaction: Rate slowest for touch-screen interface with dynamic display (see Fig. 5.1).

Predictions for simple effects based on predicted interaction:
dynamic>static with touch-screen interface

joystick>touch-screen with static display

dynamic will not differ significantly from static with joystick interface

joystick will not differ significantly from touch-screen with dynamic display

Commented Program for Design 7 (Program File Name: design7.sps)

This program has separate parts for Factor 1 and Factor 2. The output appears for each factor as that part of the program is completed. This device is necessary because the data have to be correctly ordered before each part, and sorting must be done outside the matrix language. Notes are provided only for the first part. In the second part the two factors reverse roles.

```
set mxloops 5000.                          increase the maximum loop size to 5000
sort cases by factor2(A).                  arrange the data according to levels of factor 2
matrix.                                    start the matrix language
get limits/variables=limits/missing=omit.  form a matrix from the first column of the data window
get data/variables data factor1 factor2/missing omit.
                                           form a matrix from the second, third, and fourth columns of the data window
compute ncase=limits(1).                   find the number of rows of data
compute nreps=limits(1)/4.                 find the number of replicates
compute nswaps=Nmits(1)/2.                 find the number of observations at each level of factor 2 (for rearranging within levels of factor 2)
compute totalf1={0,0}.                     start factor 1 level totals at zero
loop case=1 to ncase.                      and collect the factor 1 level totals for the actual data
compute totalf1 (data(case,2))=totalf1 (data(case,2))+data(case,1).
end loop.
compute test1=(totalf1(2)−totalf1(1))/(nreps*2).
print test1/title="factor 1 test statistic".  and the test statistic for factor 1
compute nperm=2001.                        this is the number of arrangements +1 for the actual data
```

```
compute results1=uniform(nperm,1).          make a matrix of the correct
                                            shape to receive the results—it
                                            is full of random numbers but
                                            will be overwritten later
compute results1 (1,1)=test1−test1/1000000.
                                            put the actual test statistic in the
                                            first place in the results matrix,
                                            reduced by a very small
                                            multiple of itself to avoid
                                            comparison problems
compute pos1=0.                             this will be the count of
                                            arrangement statistics
                                            at least as large as the actual test
                                            statistic
compute pos2=0.                             this will be the count of
                                            arrangement statistics
                                            at least as large in absolute value
                                              as the absolute value of the
                                            actual test statistic
loop perm=2 to nperm.                       now start the rearrangements
loop fac2=1 to 2.                           (the first is just the actual data)
loop n=1 to nswaps.                         these loops shuffle the data
                                            within levels of factor 2
compute k=trunc(uniform(1,1)*(nswaps−n+1))+n+nswaps*(fac2–1).
compute case=n+nswaps*(fac2–1).
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
end loop.
compute totalf1={0,0}.                      start factor 1 level totals at zero
loop case=1 to ncase.                       collect factor 1 level totals for
                                            this arrangement
compute totalf1 (data(case,2))=totalf1 (data(case,2))+data(case,1).
end loop,
compute test1=totalf1 (2)−totalf1(1))/(nreps*2).
                                            and the test statistic
compute results 1 (perm,1)=test1.           and put in the results matrix
end loop.                                   next arrangement
compute absres1=abs(results1).              find absolute values for
                                            two-tailed test
loop k=2 to nperm.                          now compare arrangement test
                                            statistics with the actual one, and
                                            count those at least as large
```

```
do if results1 (k,1)>=results1 (1,1).
compute pos1=pos1+1.                          and for absolute values
end if.
do if absres1 (k,1)>=absres1 (1,1).
compute pos2=pos2+1.
end If.
end loop.
print pos1/title="count of arrangement statistics at least as large",
compute prob1=pos1+1)/nperm.                  calculate the one-tailed
                                              probability
print prob1/title="factor 1 one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(tes
compute prob2=(pos2+1)/nperm.                 and the two-tailed probability
print prob2/title="factor 1 two tail probability".
end matrix.                                   end of the matrix language
sort cases by factor1 (A).                    arrange the data according to
                                              levels of factor 1
matrix.                                       restart the matrix language for
                                              part 2
get limits/variables=limits/missing=omit.
get data/variables data factor1 factor2/missing omit.
compute ncase=limits(1).
compute nreps=limits(1)/4.
compute nswaps=limits(1)/2.
compute totalf2={0,0}.
loop case=1 to ncase.
compute totalf2(data(case,3))=totalf2(data(case,3))+data(case,1).
end loop.
compute test2=(totalf2(2)−totalf2(1))/(nreps*2).
print test2/title="factor 2 test statistic".
compute nperm=2001.
compute results2=uniform(nperm,1).
compute results2(1,1)=test2−test2/1000000.
compute pos1=0.
compute pos2=0.
loop perm=2 to nperm.
loop fac1=1 to 2.
loop n= 1 to nswaps.
compute k=trunc(uniform(1,1)*(nswaps−n+1))+n+nswaps*(fac1−1).
compute case=n+nswaps*(fac1−1).
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
end loop.
compute totalf2={0,0}.
loop case=1 to ncase.
```

```
compute totalf2(data(case,3))=totalf2(data(case,3))+data(case,1).
end loop.
compute test2=(totalf2(2)−totalf2(1))/(nreps*2).
compute results2(perm,1)=test2.
end loop.
compute absres2=abs(results2).
loop k=2 to nperm.
do if results2(k,1)>=results2(1,1).
compute pos1 =pos1 +1.
end if.
do if absres2(k,1)>=absres2(1,1).
compute pos2=pos2+1.
end if.
end loop.
print pos1/title="count of arrangement statistics at least as large".
compute prob1=(pos1+1)/nperm.
print prob1/title="factor 2 one tail probability".
print pos2/title="count of arrangement statistics at least as large in abs value as abs(tes1
compute prob2=(pos2+1)/nperm.
print prob2/title="factor 2 two tail probability".
end matrix.
```

Randomization Test Results for Design 7 Example

Output	*1st run*	*2nd run*	*3rd run*
Main Effect for Factor 1 (interface device)			
Test statistic	1.000	1.000	1.000
No. as large	234	221	214
One-tailed probability	0.117	0.111	0.107
No. as large in absolute value	461	435	427
Two-tailed probability	0.231	0.218	0.214
Main Effect for Factor 2 (display mode)			
Test statistic	1.500	1.500	1.500
No. as large	63	68	67
One-tailed probability	0.032	0.034	0.034
No. as large in absolute value	134	152	140
Two-tailed probability	0.067	0.076	0.070

Mean time for three runs=27 sec

Statistical Conclusions for Design 7 (One-Tailed Main Effects) Example

Randomization tests of the main effects in a 2×2 factorial experiment on a single communication aid user were carried out. In a test of the prediction that rate of communication would be faster when the interface device was a joystick rather than a touch-screen, the

proportion of 2000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.117. Therefore, the main effect of interface device was not statistically significant ($p>0.05$; one-tailed). In a test of the prediction that rate of communication would be faster when a dynamic rather than a static display mode was used, the proportion of 2000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.032. Therefore, the main effect of display mode was statistically significant ($p<0.05$; one-tailed).

Testing Simple Effects in the Factorial Design

Although there is no randomization test available for testing the interaction between interface device and display mode, if a particular form of interaction has been predicted, predictions for simple effects can be derived from the predicted interaction, and these can be tested using a randomization test. These are, of course, precisely the follow-up tests that would normally be made following the finding of a significant interaction. The tests of simple effects can be carried out using the program for Design 5a. We present the results for a randomization test of one of the four simple effects, that predicting that a dynamic display mode will be superior to a static display mode when a touch-screen interface is used.

The data for entry into the Design 5a worksheet would be as follows. At the top of Column 1, the number of observations involving only the touch-screen interface (i.e., 8) is entered. Column 2 will contain the touch-screen data for static and dynamic display modes and the display mode codes will be entered in Column 3, where the display mode predicted to result in a faster communication rate (i.e., the dynamic mode) is coded 2. The worksheet entries are shown in the Design 5a Worksheet Box (Simple Effects Example) under "Example for Design 7" in chapter 5. The file name of the worksheet on the CD-ROM is *simple.sav*.

Randomization Test Results for Design 7 (Simple Effect of Display Mode With Touch-Screen Interface) Example

Output	*1st run*	*2nd run*	*3rd run*
Test statistic	3.250	3.250	3.250
No. as large	25	23	21
One-tailed probability	0.013	0.012	0.011
No. as large in absolute value	54	44	48
Two-tailed probability	0.027	0.022	0.0024

Mean time for three runs=22 sec

Statistical Conclusion for One-Tailed Test of a Simple Effect

Predictions for simple effects were derived from the predicted form of the interaction between interface device and display mode. In a randomization test of the prediction that a dynamic display mode would result in a faster communication rate than a static dis-

play mode when the interface device was a touch-screen, the proportion of 2000 randomly sampled data divisions giving a difference in the predicted direction at least as large as the experimentally obtained difference was 0.013. Therefore, the simple effect of dynamic display with a touchscreen interface was statistically significant ($p<0.05$; one-tailed).

DESIGN 8 (ORDINAL PREDICTIONS WITHIN NONEXPERIMENTAL DESIGNS)

The randomization test that we present for application to a range of designs involving ordinal predictions is only valid in a strict sense when a genuine random assignment procedure has been incorporated in the design. We believe that there may be some circumstances in which use of the randomization test is justified even though a genuinely random procedure for the assignment of experimental units to observation periods has not been followed. We trust that those interested in using the test will read our discussion of the issue in chapter 10 and come to a view of the appropriateness of the test for analysis of their own data.

Example data for a predicted unique order and for a predicted partial order were presented in chapter 5. Each prediction will be tested in a separate run of the program. The file names of the worksheets on the CD-ROM for the unique and partial predictions are *design8uniq.sav* and *design8part.sav,* respectively.

Specifications for Design 8 Example

Number of participants (or number of observations on a single participant)	=	6
Predicted order: Unique or partial		

Commented Program for Design 8 (Program File Name: design8.sps)

```
set mxloop=5000.                                  increase the maximum loop size to 5000

matrix.                                           start the matrix language
get limits/variables=limits/missing=omit.         form a matrix from the first column of the data window

get data/variables=data predict/missing=omit.     form a matrix from the second and third columns of the data window

compute ncase=limits(1).                          collect the number of rows of data

compute sumprod=0.                                collect the sum of products for the actual data

loop case=1 to ncase.
compute sumprod=sumprod+data(case,1)*data(case,2).
end loop.
print sumprod/title="sum of products".
```

```
compute nperm=2001.                                  this is the number of
                                                     arrangements+ for the actual
                                                     data
compute results=uniform(nperm,1).                    make a matrix of the correct
                                                     shape to receive the results—it
                                                     is full of random numbers but
                                                     will be overwritten later
compute results(1,1)=sumprod−sumprod/1000000.
                                                     put the actual test statistic in the
                                                     first place in the results matrix,
                                                     decreased by a very small
                                                     multiple of itself to avoid
                                                     comparison problems
compute pos=0.                                       this will be the count of
                                                     arrangement statistics
                                                     at least as large as the actual
                                                     test statistic
loop perm=2 to nperm.                                now start the rearrangements
loop case=1 to ncase.                                (the first is just the actual data)
                                                     this loop shuffles the data
compute k=trunc(uniform(1,1)*(ncase−case+1))+case.
compute temp=data(case,1).
compute data(case,1)=data(k,1).
compute data(k,1)=temp.
end loop.
compute sumprod=0.                                   collect the sum of products for
                                                     this arrangement
loop case=1 to ncase.
compute sumprod=sumprod+data(case,1)*data(case,2).
end loop.
compute results(perm,1)=sumprod.                     and put in the results matrix
end loop.                                            next arrangement
loop k=2 to nperm.                                   now compare arrangement test
                                                     statistics with the actual one, and
                                                     count those at least as large
do if results(k,1)>=results(1,1).
compute pos=pos+1.
end if.
end loop.
print pos/title="count of arrangement sums of products at least as large".
compute prob=(pos+1)/nperm.                          calculate the probability
print prob/title="probability".
end matrix.                                          end of the matrix language
```

Randomization Test Results for Design 8 Example

Output	*1st run*	*2nd run*	*3rd run*
	Test of the Unique Order Prediction		
Test statistic	110.000	110.000	110.000
No. as large	35	38	31
One-tailed probability	0.018	0.019	0.016
	Test of the Partial Order Prediction		
Test statistic	44.000	44.000	44.000
No. as large	198	205	195
One-tailed probability	0.099	0.103	0.098

Mean time for six runs=13 sec

Statistical Conclusion for Design 8 (Unique Order Prediction of Small Group Data) Example

In a randomization test of the prediction of a unique ordering of communicative competence ratings of communication aid users, the proportion of 2000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.018. Therefore, the correlation between the obtained order and the predicted order (based on the etiology of speech impairment and severity of additional physical impairments) was statistically significant ($p<0.05$; one-tailed).

Statistical Conclusion for Design 8 (Partial Order Prediction of Single-Case Data) Example

In a randomization test of the prediction of a partial ordering of frequency of questions asked by a communication aid user of same-sex and opposite-sex partners, the proportion of 2000 randomly sampled data divisions of obtained and predicted orders giving a statistic at least as large as the experimentally obtained (sum of products) statistic was 0.099. Therefore, the correlation between the obtained order and the predicted order (based on the gender of partners) was not statistically significant ($p>0.05$; one-tailed).

Chapter 9
Other Sources of Randomization Tests

In chapters 6 to 8 we provided macros to run within three different packages to carry out randomization tests on the designs we listed in chapter 5. There are various alternative sources of descriptions of experimental designs suitable for analysis using randomization tests and software for carrying out such tests. Some are distributed in books and journal articles, some are in statistical packages dedicated to tests of this kind, and some are included in well-known general statistical packages. Although we know of only one statistical package specifically designed for use with single-case data, most packages containing randomization tests include some that can be applied to those single-case designs for which analogous group designs exist. In this chapter, we provide a brief, nonexhaustive summary of sources. In the case of journal articles and books, apart from those that we have had occasion to cite, we confine ourselves to providing details of a bibliography that is available via e-mail or World Wide Web. In the case of packages, we indicate features that seem to us to relate to their usefulness for the purpose of making statistical inferences about data from single-case and small-*n* designs, but we do not attempt anything approaching systematic evaluation of the packages.

BOOKS AND JOURNAL ARTICLES

There is an excellent EXACT-STATS mailing list (randomization tests are often referred to as exact tests). It is not necessary to join the list to access the bibliography. The bibliography is organized in four sections:

(A) concepts and designs (subdivided into books and articles), (B) algorithms, (C) preprints and recent publications (books and articles), and (D) comprehensive listing (i.e., of publications on exact tests and related topics). Sections of the bibliography can be obtained free by e-mail or via World Wide Web. To obtain the Section A bibliography; send the e-mail message:

send exact-stats bibliogen.txt
to the address:
mailbase@mailbase.ac.uk
or visit:
http://www.mailbase.ac.uk/lists/exact-stats/files/bibliogen.txt
To obtain other sections of the bibliography, replace bibliogen.txt in the preceding addresses with:
biblioalg.txt (for Section B)
bibliopre.txt (for Section C)
bibliocom.txt (for Section D)

STATISTICAL PACKAGES

We do not aim to provide an exhaustive list of packages containing procedures for randomization (exact) tests. Rather, we provide brief information on a sample that we have direct experience with. A more extensive list of sources is provided by Baker (1995) in an appendix to Edgington's (1995) book *Randomization Tests* (3rd ed.).

RANDIBM

This is a package of randomization tests presented by Edgington (1995) in his book *Randomization Tests* (3rd ed.), which are best used in conjunction with the discussion of the tests provided in the book. They can be used on IBM-compatible computers to analyze data from several group and single-case designs and can be obtained free via anonymous file transfer protocol. The link between the software and the detailed discussion of the procedures in the book is an attractive feature. On the negative side, accessing the package via DOS and entering data within DOS may be unfamiliar procedures for many potential users. We suspect that PC users who are not computer enthusiasts and were "brought up" on a Windows environment or who converted to Windows some years ago, may find the DOS environment offputting. A limitation of this package of tests is that it contains no programs for the analysis of phase designs.

To access the package, you will need to obtain a DOS prompt. When the DOS prompt is displayed (e.g., C:\), the following commands should be used to download RANDIBM, where system prompts are shown in normal typeface and user entries are shown in bold:

C:\>**ftp ftp.acs.ucalgary.ca**
User: **anonymous**
Password: **guest**
ftp>**cd pub/private_groupjnfo/randibm**
ftp>**get readme.doc**
ftp>**binary**
ftp>**get randibm.exe**
ftp>bye

The two files, readme.doc and randibm.exe will (with the DOS prompt as in the given example) be downloaded to C:\. The readme.doc file can be opened in Microsoft Word and the randibm.exe file can be run by double clicking on the filename in the C:\ directory in Windows. When asked to select "Color or Monochrome monitor," type *M* to select monochrome.

SCRT

This is the Single-Case Randomization Tests (SCRT) package developed by Onghena and Van Damme (1994). It is the only comprehensive package that we know of that was developed

specifically for single-case designs. It is available from iec ProGAMMA, P.O. Box 841, 9700 AV Groningen, The Netherlands, at a cost of U.S. $375 (or U.S. $250 educational price). As with Edgington's (1995) RANDIBM package, it suffers from the disadvantage of running under DOS. However, once into the package, the interface is reasonably friendly, if a little cluttered, and the mouse can be used to select from the various menus and to move around the screen. Nonetheless, having become used to entering data in Windows applications, including importing whole data files, the data entry in this DOS package does seem rather laborious.

The most attractive feature of the package is the facility to build and analyze a range of tailor-made randomization designs, and the option of using systematic or randomly sampled arrangements of the data for each design. The manual is concise and reasonably helpful provided the reader is already familiar with a range of single-case experimental designs.

StatXact

This is the most comprehensive package available for the application of randomization procedures to a wide range of nonparametric statistics for which probabilities are normally based on the theoretical sampling distributions of the test statistics. It is available from Cytel Software Corporation, 675 Massachusetts Avenue, Cambridge, MA 02139, USA, and information about the product can be accessed at *http:// www.cytel.com.* StatXact 4 for Windows costs U.S. $955 for a single user at the time of writing.

We are familiar with StatXact 3 and have used it to carry out randomization tests on single-case data from designs that are analogous to group designs that can be analyzed using Kruskal-Wallis, Mann-Whitney, and Wilcoxon tests, respectively (i.e., equivalent to our Designs 5, 5a, and 6a). Various alternative analyses are provided for these designs in StatXact, mostly using ranked data. There are options, however, for carrying out analyses on the raw data for the three designs just referred to. The option for Design 5 is called ANOVA with General Scores, and the options for Designs 5a and 6a are both called Permutation. These options yield, within the limits of random sampling error, the same probabilities as our macros. Strangely, there is no equivalent option for a design that can be analyzed using a Friedman test (i.e., equivalent to our Design 5). Most of the alternative StatXact 3 analyses available for our Designs 5, 5a, and 6a are special cases of the raw data analyses and, as such, are likely to have only rather specialized applications. Either exact probabilities (i.e., based on systematic generation of all possible arrangements) or Monte Carlo probabilities (i.e., based on random sampling from all possible arrangements) can be selected.

A limitation of StatXact is that it lacks analyses for single-case designs for which there are no analogous group designs with associated nonparametric analyses and it lacks facilities for building such analyses. This view was echoed in a generally favorable review of StatXact 3 by Onghena and Van Den Noortgate (1997). They suggested that, "Instead of increasing the number of ready-to-use procedures, it could be worthwhile to have a general permutation facility that makes it easy to devise one's own new procedures, even with unconventional test statistics" (p. 372). On their Web site, Cytel claim that they have done this in StatXact 4. They say, "We've provided the tools for you to construct your own procedures," but we were unable to find any further details about this on the Web site. Also, there was no indication in the StatXact 4 "road map" that procedures for single-case designs had been added.

The StatXact manual is technically comprehensive, although somewhat daunting. Our feeling, overall, is that this package will appeal to statistics aficionados. As most of the commonly required procedures are available free or at much lower cost, or as add-ons to major gen-eral-purpose statistical packages, such as SPSS for Windows, its appeal for the average clinical researcher looking for a quick statistical solution may be limited.

SPSS for Windows

This widely used general statistical package includes an Exact Tests Module, that provides exact probabilities for a range of nonparametric test statistics. As with StatXact, either exact probabilities or Monte Carlo probabilities can be selected. The exact tests corresponding to those we provide for Designs 5, 5a, 6, and 6a are all carried out on the raw data (including the Friedman test for Design 5a) and produce the same probabilities (within the limits of random sampling error) as our macros. SPSS does not provide anything like the range of specialized procedures available in StatXact, but, for researchers with average statistical expertise, this may be advantageous. The exact test procedures that SPSS provides are the ones that are most often required, and a great deal of potential for confusion is avoided.

Like StatXact, the SPSS Exact Tests Module provides no tests for single-case designs for which there is no group design analogue. In chapter 8, we provided randomization tests to run within SPSS for single-case designs both with and without group design analogues. Although those with group design analogues produce the same results as the SPSS exact tests, we hope that our presentation of them in chapter 5 as group and single-case variants of the same basic designs will encourage clinical researchers to use them. However, researchers who have access to SPSS for Windows will probably also be interested in exploring the exact test facilities provided within the SPSS module. As well as our own programs, users with some expertise may also find program listings for several randomization tests provided by Hayes (1998) of interest. For anyone with an SPSS Base system, but without the SPSS Exact Tests add-on, the additional cost for a single user at the time of writing is U.S. $499, although "educational" discounts may apply. SPSS can be contacted at *http://www.spss.com.*

SAS

This is another general statistical package, that provides some facilities (procedures) to assist users in carrying out some randomization tests. Researchers who are already familiar with the package may be interested in exploring the use of built-in SAS EXACT options in procedures such as FREQ and NPAR1WAY for some exact tests with standard nonparametric statistics (including Monte Carlo sampling in the latest release). However, the NPAR1WAY > EXACT procedure does not allow an option to use raw scores, which means that the solutions would pro-vide only approximations to those we present for our randomization tests on our example data for equivalent designs (i.e., Designs 5 and 5a). In addition to built-in exact tests, Chen and Dunlap (1993) published SAS code that makes use of procedures, such as SHUFFLE, that may be helpful to experienced users of the package who are interested in constructing their own randomization tests. SAS can be contacted at SAS Institute Inc., SAS Campus Drive, Gary, NC 27513–2414, USA, or at *http://www.sas.com.*

Chapter 10
The Use of Randomization Tests With Nonrandomized Designs

We have emphasized that the validity of a randomization test rests on the way rearrangements of the data carried out in the test reflect the possible data arrangements based on a random assignment procedure adopted as part of an experimental design. When there has been no random assignment of treatments to participants or observation periods, no valid randomization test will be possible. Also, if there is a mismatch between the way in which treatments are randomly assigned and the way in which arrangements are generated in a randomization test, it will not be valid. We provide an example of how this might arise by revisiting the AB phase design (our Design 1).

In our consideration of the AB design, we took for granted that the phases have a logical order that must not be violated by any random assignment procedure. Consequently, we adopted Edgington's (1995) randomization procedure, whereby the point of intervention—the change from the A phase to the B phase—was randomly assigned within some prespecified range of observation periods. The randomization test then compared the obtained statistic (the difference between phase means) with a random sample of alternative values of the statistic that would have occurred with other possible intervention points if the treatment had no effect (i.e., the data were the same). Thus, the way in which arrangements of the data were generated matched the way in which treatments were randomly assigned.

Suppose that we had followed Levin et al.'s (1978) suggestion to abandon the logic of the phase order and to randomly order the complete phases. Suppose, also, that there were two observation periods in each phase (i.e., A1, A2, B1, B2). With the intact phase as the unit of analysis, there still would be only two possible arrangements under the random assignment procedure (i.e., AABB or BBAA). A randomization test should then compare the obtained statistic (the difference between phase means) with the value that would have been found with the single alternative arrangement that was possible. The randomization test would, of course, be pointless because the p value could not be less than 0.5. It would, however, have maintained a match between the random assignment procedure and the generation of alternative data arrangements in the test.

Suppose, on the other hand, that we carried out a randomization test in which all possible arrangements of individual observations from the two phases were generated. The additional arrangements generated in the randomization test (i.e., ABAB, ABBA, BABA, and BAAB) could not have occurred in the random assignment of treatments to phases, so there would be a mismatch between the random assignment procedure and the generation of arrangements in the test. In this case the test might lead to the conclusion that the probability of the obtained difference between treatment means being the largest difference was 1/6=0.17, whereas in reality the probability was ½=0.50. This is analogous to the situation described by Campbell and Stanley (1966), in which treatments are randomly assigned to intact classrooms but individual children are treated, erroneously, as the experimental units in the statistical analysis.

One of the attractions of the randomization test approach is the way in which the form of random assignment in the design directly determines the set of possible data arrangements under the null hypothesis. Strictly, if this link is violated any statistical inference will be invalid. In large-*n* studies, however, it is often considered acceptable to relax the strict requirements of statistical tests provided that caution is exercised in the interpretation of resulting probabilities. It is not unusual, for example, for normality and variance assumptions required by parametric tests to be relaxed in the name of robustness of the test statistic, and the requirement for random sampling is hardly ever met in human research. More directly analogous to the randomization test situation is the violation of the requirement for random assignment of treatments to participants, which is common in large-*n* designs. For example, classification variables, such as gender, age, and IQ, are frequently included as independent variables in group studies. There is, of course, no question of levels of these variables being randomly assigned to participants by the researcher. As Minium, King, and Bear (1993) made clear, studies lacking random assignment of the independent variable cannot correctly be called experiments. When a statistically significant result is found, we can conclude that there was probably some systematic effect on the dependent variable, but we cannot safely infer that it was an effect of the classification variable because different levels of that variable (e.g., biological gender) are likely to be associated with differences on other variables (e.g., socialization practices). As Minium et al. also made clear, however, this does not mean that nothing useful can be learned from designs using classification variables.

We are certainly not advocating that the randomization requirement be treated lightly, but we believe there is no sound reason that, with the same caveat regarding caution in the interpretation of statistical probabilities, similar relaxation of the random assignment requirement in a single-case (or very small-*n*) study should be prohibited in all circumstances. Of course a random assignment procedure should be used whenever it is practical. For example, we see little justification for failing to select an intervention point randomly in an AB phase design when the intended experimental units are individual observations. This is particularly the case because stability can be established in prebaseline trials if that kind of responsive procedure is considered important (see Ferron & Ware, 1994, for a useful discussion of the general issue of randomized vs. responsive designs). Nonetheless, there are circumstances in which a randomization test applied to a nonexperimental design (i.e., one lacking a random assignment procedure) may be preferable to using a less satisfactory test or none at all. Edgington (1995) suggested that use of a randomization test in this way may help a researcher to decide whether it is likely to be worth continuing with investigations of the variables. Again, Levin et al. (1978) argued quite strongly that the application of randomization tests in the case of systematic, as opposed to random, assignment of units to phases may be viewed as an appropriate approximation.

NONRANDOMIZED DESIGNS

Nonrandomized Classification Variables

There are, in particular, two related kinds of circumstance in which we think there may be a case for using a randomization test in the absence of a random assignment procedure in

the design. These circumstances are related conceptually in that they both come down to testing ordinal predictions, and they are related practically in that they can both be dealt with using the same macro. One is the situation in which ordinal predictions are made for a dependent variable on the basis of preexisting characteristics of individuals or the environment (i.e., classification variables). This is the situation that would correspond to our Design 8 if the experimental units were not assigned randomly to observation pe-riods, and it is directly comparable to the use of classification variables in large group studies, which is by no means rare. Examples for very small-*n* and single-case designs were presented under "Example for Design 8" in chapter 5. If the participants or, for the single-case example, the participant's conversational partners, were assigned at random to the six available observation periods, the randomization test would be valid. If, however, they were assigned in an arbitrary way, based on availability, the test would not be statistically valid. Nonetheless, a truly arbitrary assignment might be taken as a fair approximation to a random procedure and the test might still provide useful information. Had they been assigned in a systematic way (e.g., ordered according to their status on the independent variables), it would be harder to justify use of the randomization test. Researchers need to consider the specifics of their assignment procedure to decide on the plausibility of treating arbitrary assignment as a substitute for random assignment.

Nonrcmdomized Phase Designs With Specific Predictions

The second circumstance in which we might be inclined to relax the random assignment requirement arises occasionally within designs for which the logic of the design requires a systematic sequence of phases, as in the AB design. It has been argued (e.g., Onghena, 1992)—and we concur—that the valid randomization procedure for phase designs is to order the phases as logic dictates, then to randomly assign the intervention point within some predetermined range of observations (i.e., as in our Design 1). The same reasoning applies to extensions of the basic AB design (e.g., our Designs 2–4 and ABAB designs).

As we noted in our earlier discussion of the AB design, Levin et al. (1978) advocated a different procedure, that of randomly assigning treatments to whole phases and basing the randomization test on all possible arrangements of phases. They elaborated on this approach particularly with respect to the ABAB design, and that is the design that we take as our example. With treatments randomly assigned to four phases, the design is correctly classified as a randomized treatment (or alternating) design (i.e., our single-case version of Design 5) rather than an ABAB phase design. We may, however, pursue Levin et al.'s approach a little further, avoiding inclusion of possible random orderings of phases that are incompatible with an ABAB design, such as an AABB sequence, which would effectively constitute an AB design. Of course, if the ABAB sequence is retained, assignment of treatments to phases will be, necessarily, systematic rather than random.

Apart from the absence of a random assignment procedure, there are, as Onghena (1992) pointed out, additional problems with this approach. In the first place, if phases are used as the experimental units, the analysis will lack power because the maximum possible number of arrangements of phases is six (i.e., the smallest possible p value=1/6= 0.167). However, as we see in the next chapter, the power of randomization tests applied to phase designs with random assignment of intervention and withdrawal points is not impressive either.

Ferron and Onghena (1996) suggested that one solution may be to combine both kinds of randomization (i.e., random selection of transition points and random assignment of treatments to the resulting phases) within a single design, but this would still violate the logical order implied by an ABAB design. We have some sympathy, therefore, with Levin et al.'s (1978) willingness to contemplate systematic assignment of treatments to ABAB phases, when power can be increased by making more specific predictions. If, for example, the prediction, B2>B1>A2>A1, can be made, there are 4!=24 possible ordinal arrangements of the data and the smallest possible p value is 0.042. As the predicted order of the phase means does not correspond to the ordering of the phases, the plausibility of practice (or fatigue) effects accounting for the order of means when the hypothesis is supported is limited.

It may be noted that the downside of making specific predictions is that it is a somewhat "rigid" strategy. In the preceding example, the hypothesis would only receive support at $p<0.05$ if the exact predicted order was the one obtained. This is not really different in principle, however, from making a one-tailed prediction to increase the sensitivity of a test. As in that situation, it should hardly be necessary to emphasize that the predicted order must be generated before data are obtained.

The application of a randomization test to ordinal predictions of this kind can again be dealt with by the macros provided for our Design 8. All that is required to test the prediction B2>B1>A1>A1 is to specify the order, 1, 3, 2, 4 to correspond to the means for phases A1, B1, A2, B2 in the worksheet (see chap. 5). It should be clear that predictions for partial orderings can also be accommodated within Design 8. Thus, for example, the prediction (B1 and B2)>A2>A1 would require the specification of the order 1, 3, 2, 3 to correspond to the phase means A1, B1, A2, B2. It is also straightforward to test stronger predictions involving different degrees of difference between phases. Suppose, for example, that it was predicted that the order of phase means would be B2>B1>A2>A1 and, additionally, that differences between baseline (B) phases and treatment (A) phases would be greater than the differences between phases of the same kind (e.g., A1, and A2). Then, the required interval specification could be 1, 4, 2, 5 (or 1, 6, 2, 7, etc., depending on how big the differences across phase types and within phase types were predicted to be).

Marascuilo and Busk (1988), in their discussion of ABAB designs with the phase as the unit of analysis, presented the randomization tests we have described here as examples of trend tests based on coefficients for linear contrasts derived from orthogonal polynomials. This is per-fectly correct, but it is unnecessary for a researcher to be familiar with the theory of linear contrasts to be able to use our macros for Design 8. We believe that our presentation of the test as a correlation between predicted and obtained orderings will seem more intuitively straightforward, and that the ease of use of the macros will encourage researchers to experiment more than they have thus far.

There may be other circumstances when it is reasonable to relax the random assignment requirement for carrying out a randomization test. It may, for example, be a sensible strategy for the analysis of pilot data or the analysis of existing data, where the analysis implications of failing to incorporate a random assignment procedure were not realized when the data were collected. The analysis solution may not be ideal, but it still may be preferable to the alternatives. In every case, the key consideration should be realism about the extent to which the statistical conclusion provides support for a causal hypothesis, and this should be reflected in the caution with which the results are reported.

Chapter 11
The Power of Randomization Tests

Consideration of the power of classical statistical tests lagged behind the concern with *p* values. So it has been with randomization tests. For the sake of readers who are hazy about the concept of power, we begin with a brief summary of how power relates to other concepts we have discussed, before going on to take a look at what is known about the power of randomization tests. For readers who want a more detailed treatment of power, Allison, Silverstein, and Gorman (1996) is both clear and practically useful.

THE CONCEPT OF POWER

There are two possible reality states regarding the null hypothesis. Either it is true (there is not an effect) or it is false (there is an effect). There are also two possible decision outcomes following a statistical test of the null hypothesis. The decision is either to reject it (a significant effect is found) or to accept it (a significant effect is not found), where accept is understood to stand for fail to reject. That means that all inferential possibilities can be represented in a two-by-two table as illustrated in Fig. 11.1.

We have presented the information in Fig. 11.1 in terms of whether there is or is not an effect in reality and whether or not an effect is found by the statistical test, rather than in terms of the null hypothesis, because we think the former is intuitively more straightforward.

We see from Fig. 11.1 that *power* refers to the sensitivity of a test to an effect that exists in reality. It is the probability of correctly inferring that an effect exists We see, also, that

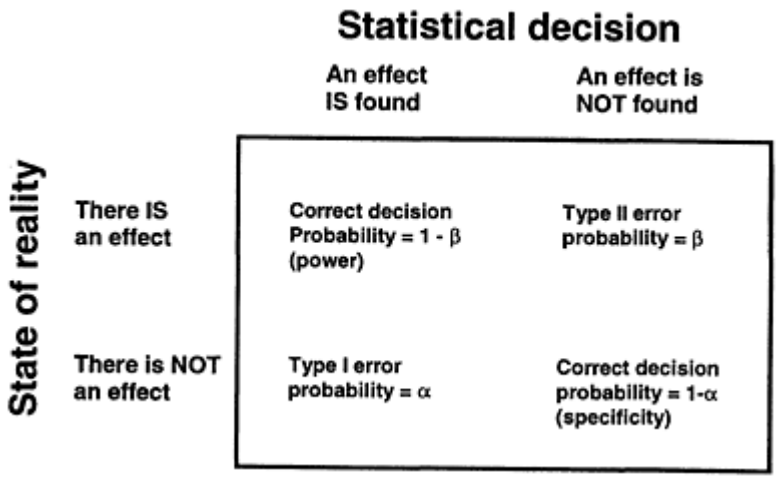

FIG. 11.1. The probabilities of inferential outcomes following a statistical test.

the probability of missing a real effect is denoted by the symbol β and that this outcome is referred to as a *Type II error*. If the probability of making a Type II error were 0.2, the power of the test would be 0.8 (i.e., $1-\beta$). In the past, far less attention has been paid to power and Type II errors—the other side of the power coin—than has been paid to Type I errors. Researchers routinely set an α level (the probability of finding an effect that is not real), typically at α=0.05 or 0.01, to limit the chance of erroneously inferring an effect. The general assumption is that this kind of error is more serious than the kind that results in failure to find a real effect. Thus, setting a β level to limit the chance of missing an effect is a relatively recent innovation, and the risk that is implied by the level that is considered acceptable for β (typically β=0.2 or 0.1; power=0.8 or 0.9) is much higher than that implied by the level considered acceptable for α.

Although power considerations have recently become more prevalent in large-*n* designs, it is still the case that rather little attention has been paid to the issue with respect to single-case experiments. This may have been, in part, because of the less strong tradition of statistical analysis in single-case research. It may also have been partly an incidental consequence of the emphasis that has been placed on the dependence of power on the number *(n)* of participants in a design, which clearly does not apply in the case of $n=1$ designs. It has been suggested (e.g., Franklin, Allison, et al., 1996) that number of observations affects power in single-case studies in the same way as number of participants does in large-*n* studies, but, as we shall see, this is an oversimplification.

There is also a principled objection to the use of quantitative power determination in the absence of random sampling, which always applies for single-case experiments and, realistically, for most group experiments as well (E.S.Edgington, personal communication, April 4, 2000). Our own view is that the theoretical objection is well founded but, as the procedures are widely accepted for group designs, their omission in the treatment of Randomization Tests for single-case designs is likely to be construed as a weakness of the randomization approach. Moreover, we think that when power determination procedures are stripped of their spurious precision, they provide useful, approximate guidelines.

THE DETERMINANTS OF POWER

Before considering the power of randomization tests, we summarize the widely accepted views about power as it applies to large-*n* designs. Much of this applies equally to randomization tests, but for these tests there are some additional considerations about which there is less consensus.

The Probability of Type I Error (α Level)

This is directly under the researcher's control. The lower the α level is set, the less likely a Type I error; that is, the less likely it will be that a significant effect will be found, mistakenly, when the null hypothesis is true. If there is a low probability of finding an effect by mistake, this will imply a relatively high probability of missing a real effect (i.e., a Type II error). That means that a low value for α goes with a high value for β. As power is equal

to 1−β, a low value for a will tend to go with low power. In other words, if a researcher sets α at 0.01 rather than 0.05 to minimize false positive decisions, the power to detect true effects will be relatively low. So, power can be increased by setting a higher value for α. This can be viewed as a trade-off between the perceived importance of avoiding errors of the two types (i.e., false positives and false negatives). It is a judgment that has to be made by a researcher in light of a wide range of considerations, such as impact on theory, practical importance of outcomes, ethical issues, and cost efficiency. It may be thought that, because researchers often do not in fact set α levels at the outset, but wait until the analysis is completed to see what is the lowest *p* value they can report, there is no need to make the judgment referred to here. In the conventional view of power, this is a mistake that arises from the neglect of power considerations in the design of experiments. Any estimates concerned with power can only be made for a specified level of α. It must be conceded, however, that the reliance of power computations on fixed α levels is considered by some to be unrealistic (E.S. Edgington, personal communication, April 4, 2000). According to this view, the report of a conventional significance level reached without any α level having been set in advance simply provides a rough indication of how small a *p* value was obtained. This interpretation of conventional significance levels probably accords better with how researchers generally use them than the view that requires them to be set in advance. Nonetheless, we believe that the formal position on preset α levels does have the merit of enabling useful approximations to be obtained from power computations. Furthermore, the emphasis on a predetermined α level is reduced when, as is increasingly done, power tables containing estimates for a range of α levels (and effect sizes) are generated rather than a definitive α level being set in advance.

Effect Size

It should be obvious that, if we want to specify the probability of not missing a real effect (i.e., power), that probability must depend on how big the effect is. Big effects (e.g., big changes in means or high correlations) will be easier to avoid missing than small effects. The difficult issue is how to specify the size of effect for which we want to know the power of a test to detect. If we really knew the effect size, there would be no need to do the experiment. We may, nonetheless, have enough information, for example, based on previous research with similar variables, to enable us to make a rough guess. Alternatively, particularly in many single-case experiments, we may be able to specify an effect size that would represent clinical significance. That would be the level of effect size that we would not want to risk missing. The main point to be made here is that specifying an effect size for power calculations is a very approximate business. This reinforces our previous conclusion, in connection with preset a levels, that power calculations can only be taken as a rough guide to what is reasonable. It is for this reason that many researchers feel comfortable with the very crude guide to what constitutes large, medium, and small effect sizes (respectively, 0.8, 0.5, and 0.2 of a standard deviation difference between means) suggested by Cohen (1988).

There are various ways of measuring effect size and we do not intend to provide details of each. We recommend the chapter by Allison et al. (1996) to readers who want a clear and informative introduction. We do, however, define the commonly used measure of effect

size referred to here to aid interpretation of the large, medium, and small values sug-gested by Cohen. Effect size may be thought of in terms of the distance between means of levels of an independent variable, in some kind of standardized units so that the distance does not depend on the measurement scale used. The measure referred to by Cohen expresses the distance in units of standard deviation. Thus:

effect size=(mean of Population 1−mean of Population 2)/SD

where *SD* refers to the standard deviation of either population, as they are assumed to be equal. Thus, according to Cohen's (1988) rough guide, a medium effect size is one in which the difference between means is 0.5 of their common standard deviation.

Sample Size

In large-*n* experiments, sample size is given by the number of participants included in the study. Other things being equal, the more participants in the sample, the greater the power of a statistical test. This is a well-known statistical and empirical conclusion and, if the reader accepts it, it is not necessary to be concerned with the statistical reasoning underlying the assertion to proceed. For those who want a justification for the assertion, it is necessarily true because the variance of the sampling distribution of the mean (σ^2_{mean}) decreases as sample size *(n)* increases ($\sigma^2_{mean}=\sigma^{2/n}$). The less samples vary among themselves, the more reliable will be the mean of the sample in hand. For a more detailed explanation, the reader should consult a standard statistical text such as Howell (1997).

Other Factors Influencing Power

Power, together with the three other quantities we have listed (i.e., α level, effect size, and sample size) constitute a deterministic system, which means that if we know any three of the values, the fourth can be calculated. Before considering how this works, we mention some other ways (apart from setting α at a higher value, making *n* larger, or deciding on a larger critical effect size) in which a researcher can increase the power of an experiment.

Control of Random Nuisance Variables.

The reason we sometimes fail to find a statistically significant effect when the null hypothesis is in fact false is that there was a lot of variability between scores within the same condition. Provided that participants have been randomly assigned to conditions and treatment order, we can assume that the variability within conditions is caused by random nuisance variables. Because these random nuisance variables may happen, by chance, to favor scores in one condition, the bigger the effects of such variables, the more plausible it becomes that they are responsible for any difference between means in the two conditions. It follows that the more we can reduce the effects of random nuisance variables, the more likely we are to find a real effect of the independent variable; that is, the more powerful the test will be.

Sometimes there is an obvious nuisance variable that can be controlled by eliminating (or minimizing) its effects, as when stimuli are presented on a computer instead of by hand on flash cards, or when reinforcement is always administered by the same person

rather than by one of several people. On other occasions, it may be preferable to control a nuisance variable by making it a factor in the experimental design, as when participants are assigned to blocks on the basis of their scores on a reading test. An example of this approach applied to single-case designs would be a variant of our Design 7, in which one of the factors was a potential nuisance variable, such as time of day. When it is impractical to eliminate a random nuisance variable or to elevate it to the status of a factor in the design (e.g., because of very unequal sample sizes), it may be possible to control the nuisance variable statistically by treating it as a covariate. An ANCOVA evaluates the effect of the independent variable after allowing for any effect of the covariate. An example of statistical control of this kind in a single-case design would be the use of observation number as a covariate in a phase design. Edgington (1995) provided an illustration of how this might allow a significant treatment effect to be found even when a strong trend in the data (e.g., due to increasing boredom or fatigue) causes the means of two phases to be the same. In chapter 12, we provide an example of how the trend in Edgington's illustrative data (and other trends) can be "allowed for" in a modification of our Design 1.

When it is feasible to use repeated measures on the same participants, they act as their own control for individual difference variables. When repeated measures on the same participants are impractical (e.g., due to the likelihood of carryover effects), a less extreme form of matching participants, on the basis of a single relevant individual difference variable, may be possible. When they are the same participants in each condition, they are, of course, matched on all individual difference variables. In either case, the gain in power may be considerable when the individual differences on which participants are matched do in fact produce large effects on the dependent variable. The single-case randomized blocks version of our Design 6 is an example of the application of repeated measures to a single-case design. In this case the measures are matched within blocks of time rather than within blocks of participants. In our example in chapter 5, measures within four times of day were matched on each day of the experiment. This controls for time of day variability rather than participant variability.

There is a downside to controlling random nuisance variables, of course. They are abundant in real-world environments and controlling them always runs the risk of reducing the extrinsic (or ecological) validity of experiments. In the last resort, we are interested in the effects variables have in natural environments. However, if we fail to find a significant effect in a natural setting, that does not mean it is necessarily ineffective in that setting. It may just mean that there are other important variables as well. Increasing control over random nuisance variables increases the internal validity of an experiment—our confidence that a systematic effect can be inferred. If the effect is significant in a well-controlled experiment, we may be encouraged to explore its effect in less controlled settings. To many clinical researchers this will be familiar in terms of the statistical versus clinical significance debate. Our view is that it is a mistake to take a strong line on either internal or external validity. The constructive tension between them is such that each has a place in the research process, and the emphasis rightly shifts from one to the other in different experiments.

Increased Reliability of Measuring Instruments.

Another way of looking at the variability within conditions is in terms of the unreliability of measurements. Sometimes, variability within conditions is not due to the effects of other variables; it is just that the measuring instrument is inconsistent. If we are concerned that our bathroom scales give us different answers as we repeatedly step on and off, we assume that the problem is with the scales rather than with some other random variable. So, if we are using a rating scale to measure our dependent variable, and it has a low reliability, we would probably be wasting our time looking for random nuisance variables to control. If it is possible to improve the reliability of our measuring instrument, the power of our test should increase as the variability of measurements within a condition decreases. One way of increasing the reliability of measurement is to increase the number of measurements taken for each observation period and another is to increase the number of independent observers. In terms of ratings, we can have more raters or more ratings per rater. There is a useful chapter by Primavera, Allison, and Alfonso (1996), in which they discussed methods for promoting the reliability of measurements.

Maximizing Effect Size.

When an independent variable is operationalized, the researcher sets the degree of separation between levels of the variable. The greater the separation, the greater the size of any effect is likely to be. In our examples for Designs 1 through 4, the independent variable was availability or nonavailability of a word pre-diction system. If we had selected the best available system, we would have been likely to produce a larger effect than if we had chosen a less "state-of-the-art" system. Again in our single-case example for Design 5, had we selected only the more extreme ranges of translucency (high and low) and omitted the moderate ranges (medium/high and medium/low), we would have increased our chances of finding a large effect. Of course, when we select extreme values for our levels of the independent variable, we run the risk that the effect does not apply to most values that would be encountered in reality. Clearly there is a balance to be struck here. If we were interested in the effect of cognitive therapy on the frequency of aggressive behaviors, we might maximize any effect by providing the therapeutic intervention intensively over many months. On the other hand, if we provided only a single brief therapeutic intervention, any effect would likely be minimal. In the early stages of exploring the efficacy of a novel treatment, we might well be concerned with not missing a real effect, so increasing power by maximizing the intervention may make sense. In later research into a promising treatment, we would probably be more concerned with exploring the robustness of its effects when it is administered in a more economical way.

Researchers do not depend exclusively on their operationalization of independent variables to maximize effect size. They can produce similar effects by their choice of values for fixed features (parameters) of an experiment. For example, consider a single-case study with two randomized treatments (our Design 5a) to compare the effects of contingent and noncontingent reinforcement on frequency of positive self-statements. Choice of different lengths of observation period, different intervals between treatment occasions, or different

intervals after reinforcement sessions before commencement of observation periods would all be likely to influence effect size. For example, for alternating designs of this kind, longer "washout" intervals between treatments are likely to result in larger effect sizes.

Choice of Statistic.

It is generally held that parametric statistics are more powerful than nonparametric statistics, although it is also generally recognized that this will not always be true when assumptions required for a parametric test are not met. No general statement can be made about the relative power of randomization tests compared with their parametric and nonparametric competitors, but this is an issue that we return to later in this chapter.

Increased Precision of Prediction.

In Chapter 9 we considered the possibility of increasing the power of a test by making predictions that were more precise than the hypothesis of a difference between treatments. The increased precision arises from making ordi-nal predictions. At its simplest, this involves making a directional prediction (e.g., A>B, rather than A≠B) before data are collected, to justify using a one-tailed test. Opinion about the desirability of using one-tailed tests is divided. Our view is that this is no less acceptable than making any other designed comparison within sets of means following formulation of an a priori hypothesis. The critical requirements are that the hypothesis be formulated before data are collected and obtained differences in the nonpredicted direction result in a "not statistically significant" decision.

As indicated in chapter 9, ordinal predictions of any degree of complexity may be countenanced. The more specific the prediction, the greater the power of the test to detect that precise effect, but its power to detect near misses (e.g., a slightly different order than that predicted) becomes lower. Particularly with sparse data (which is not uncommon with single-case designs) it may be worthwhile to consider whether power could be increased by making ordinal predictions, without making the statistical decision process more rigid than seems prudent.

Estimating Power and Required Sample Size

We return now to the deterministic system comprising power, α level, effect size, and sample size. As we said earlier, if we know the values of any three, the value of the fourth can be determined. Usually, we want to know one of two things. Before conducting an experiment, we may want to know how many participants we need to ensure a specified power, given values for α and critical effect size. After conducting an experiment in which we failed to find a significant effect, we may want to know what the power of the test was, given the number of participants included in our design. Alternatively, after an experiment has been conducted, either by ourselves or by others, we may want to know the power of the test to decide how many participants to use in an experiment involving similar variables.

First, the researcher must decide on a value for α, bearing in mind that a low value for α will make it harder to achieve a high power for the test. Then the researcher must decide on a critical value for effect size. In the absence of any indications from previous related research or clinical criteria, one of Cohen's (1998) values for high, medium, or low effect size may be selected. If the intention is to determine how many participants are needed to achieve a given power, then the level of power needs to be set, bearing in mind that convention has it that a power of 0.8 is considered acceptable and a power of 0.9 is considered good. The final step is to use the values of α, effect size, and power to calculate *n,* read its value from a table or a graph, or obtain it from a computer package, bearing in mind that the value of *n* is the number of participants needed for each condition. All of these methods were discussed by Allison et al. (1996). Our view is that current packages, such as that by Borenstein, Rothstein, and Cohen (1997), have increased in versatility and ease of use to the point at which they are an extremely attractive option.

If the intention is to determine the power for a given number of participants, the α level and effect size need to be set as before, and the number *(n)* of participants per condition must be known. In this case, the final step is to use the values of α, effect size, and *n* to obtain a value for power, using any of the methods referred to earlier.

POWER IN SINGLE-CASE DESIGNS

Power functions for single-case designs are less developed than for large-*n* designs (including standard nonparametric tests). Allison et al. (1996) reported that they could find only one discussion of power in single-case research in the literature. The situation has improved since then, but the research is patchy, with large areas of uncertainty remaining. Part of the problem is that power in these designs is affected to a largely unknown degree by serial dependency (autocorrelation) in the data (see Matyas & Greenwood, 1996, for a review). This is a problem with all time-series designs (i.e., designs in which there are repeated measures on the same participants), but it is particularly problematic for single-case studies. In large-*n* studies, the effects of serial dependency can be controlled by randomizing the order of treatment administration separately for each participant. Obviously, this is not an option in single-case studies. We discussed ways of limiting autocorrelation in chapter 3. Here, we are concerned with the power consequences of failure to minimize autocorrelation.

Another complication for power considerations in single-case studies is that, starting with the same basic design, different randomization procedures are possible, each of which will have a different set of possible data arrangements associated with it. This means that the power of a randomization test will vary with the randomization procedure used. An example would be a design in which two treatments are each to be applied to two of four phases over a predetermined number of observation periods. The randomization procedure might be (a) decide on three time intervals within the sequence and randomly assign the three phase transition points within successive intervals (Ferron & Ware, 1995); (b) decide on the minimum number of observations within each phase, list all possible triplets of transition points, and randomly select one of the triplets (Onghena, 1992); or (c) randomly assign treatments to phases (Levin et al., 1978). The situation is further complicated by the

effect of number of observations per phase on the power of the test, bearing in mind that some randomization procedures, such as (b) just given, are likely to have unequal numbers of observations in each phase.

Power for a Single-Case Randomized Treatment Design

Power considerations are most straightforward for those designs that are analogues of large-*n* designs for which power functions are available. Our Design 5 (one-way small groups and single-case randomized treatment) is an example of one such design. For the single-case randomized treatment design, treatments are randomly assigned to observation times, and number of observations per treatment is analogous to number of participants per treatment in a large-*n* design, where there is random assignment of treatments to participants. Where an equivalent large-*n* design exists, the power of a randomization test can be estimated using the power function for a test that would be applied to the large-*n* design. Edgington (1969) provided an analytic proof of this, and Onghena (1994) confirmed empirically that the power for a randomization test in a single-case randomized treatment design, including when restrictions are imposed on the number of consecutive treatments of the same kind, is very close to the power of the equivalent independent groups *t* test for "group" sizes above five. For very small group sizes, the power for a *t* test is an overestimate of the power for an equivalent randomization test, although the power of the *t* test itself is very low for effect sizes in the normal range and the differences are not great. Onghena (1994) provided graphs of power functions that can be used to correct the bias, but our view is that power calculations should be regarded as very approximate estimates and the *t*-test functions will generally provide acceptable estimates for the equivalent randomization test.

Our view about the approximate nature of power calculations also bears on our decision to use random sampling of data arrangements for all of our designs (see chap. 4). We reasoned that, as well as enabling us to maintain consistency across all of the designs, this would allow the user to trade off power against time efficiency. We acknowledge that lower power is generally obtained with random (nonexhaustive) sampling from possible arrangements of the data, compared to systematic (exhaustive) generation of arrangements (Noreen, 1989), but this varies with the number of random samples taken, and the difference is reversed for some very small group sizes when sampling is saturated; that is, when the number of samples exceeds the number of possible arrangements (Onghena, 1994). Onghena and May (1995) warned against the use of random sampling when the number of observations per treatment is small (≤ 6) and, in general, when sampling is saturated. Nonetheless, we think that the general, when sampling is saturated. Nonetheless, we think that the consistency and trade-off advantages outweigh the generally small discrepancies between power functions for *t* tests and estimates for randomization tests using random (nonexhaustive) sampling, whether saturated or not.

Power for an AB Design With a Randomized Intervention Point

For single-case designs that have no large-n analogues for which power functions are available, practical guidance for choosing the number of observations required for reasonable power is sparse. Onghena (1994) drew attention to some general guidelines for maximizing

the power of randomization tests suggested by Edgington (1987)—more observations, equal numbers of observations for treatments, and more alternation possibilities between treatments—but noted that these guidelines do not include any practical suggestions for how to determine how many observations or alternation possibilities would deliver acceptable levels of power. As we observed earlier, it is very difficult to generalize about the power of randomization tests that have no analogues in classical statistics. However, an extremely useful start has been made by a few researchers.

For example, Ferron and Ware (1995) investigated the power of AB phase designs with random assignment of a treatment intervention point (our Design 1). They found low power (less than 0.5) even for an effect size as large as 1.4 with no autocorrelation present—recall that Cohen (1988) treated an effect size of 0.8 as large. With positive autocorrelation, power was even lower. Onghena (1994) obtained similar results and noted an interesting contradiction of Edgington's (1987) guideline to the effect that more observations would result in more power. This was found not to be true in all cases, as the following example shows. Suppose we have 29 observation periods with the minimum number of observations per phase set at 5. There will be 20 possible intervention points and to reject the null hypothesis at the 5% level we need the actual test statistic to be more extreme than any of the other arrangement statistics. An increase in number of observations from 29 to 39 would mean 30 possible intervention points. However, to achieve 5% significance we would still only be able to reject the null hypothesis if the actual test statistic was more extreme than any of the other arrangement statistics, because $2/30 > 0.05$. In this case our rejection region would be a smaller proportion of the sample space than in the case with only 29 observations, so power would be reduced. Of course, a would also be reduced from 0.05 to 1/30 or 0.03, but this gain would probably be of no interest, so the overall effect of increasing the number of observations from 29 to 39 would be to reduce the power, leaving the usable a level the same. Given the approximate nature of power calculations and concerns about the "unreality" of power computations based on preset α levels, however, this particular exception to Edgington's (1987) guideline concerning the relation between number of observations and power probably has limited practical importance.

Another of Edgington's (1995) guidelines found support in Onghena's (1994) simulations. This was his suggestion that power will increase with the number of alternation possibilities between treatments. In the AB design there is only one alternation possibility, and power for this design was much lower than was found for the randomized treatment design with its greater number of alternation possibilities. It was also apparent from Onghena's simulations that virtually no gain in power could be expected for the AB design as a result of making a directional prediction.

To put things into perspective, Onghena (1994) found that to achieve a power of 0.8 for an effect size of 0.8 and with a set at 0.05, an AB design with randomized assignment of the intervention point would require 10 times as many observations as a randomized treatment design. Considering the internal validity limitations of the AB design, along with its unimpressive power efficiency, it seems necessary to question the usefulness of this frequently used design. It does seem, at least, that its usefulness may be limited to very large effect sizes. This fits with the intuitions of applied behavior researchers who stress the importance of seeking large (clinically significant) effects in phase studies of this kind. It seems, however, that their statistical intuitions are mistaken when they assume that statistical analysis

of single-case phase data is likely to lead to finding too many statistically significant but clinically trivial effects. On the contrary, it seems quite likely that with the number of observations typically used, none but the largest effects will be picked up. As we observed in our discussion of Designs 1 to 4 in chapter 5, the sensitivity of phase designs with randomly assigned intervention point(s) may be increased by the addition of more intervention (e.g., reversal) points or by means of the inclusion of multiple baselines. Again, this would be consistent with Edgington's (1995) guideline concerning the relation between power and the number of alternation possibilities. Just how great the gain in power that can be achieved by such extensions is remains to be determined. If there is one general lesson that can be taken from the few studies that have addressed the issue of power in single-case designs, it is that neither number of observations nor number of possible assignments—which determines the minimum possible α value—can be used as even a rough estimate of the relative power of nonstandard designs (Ferron & Onghena, 1996).

Power for a Phase Design With Random Assignment to Phases

The preceding conclusion is consistent with the power simulations to which we now turn. These are for extensions of AB phase designs in which treatments are randomly assigned to phases rather than intervention points being randomly assigned to observation periods. These are more closely related to randomized treatment designs than to phase designs in which there is random assignment of intervention points. As Wampold and Furlong (1981) observed, the usefulness of the usual phase designs (i.e., with small numbers of phases), in which the randomization procedure involves random assignment of treatments to phases, is limited because of its lack of power, which follows from the small number of possible arrangements in, for example, an ABAB design (i.e., arrangements=6 and minimum α value=0.17). They suggested that power can be increased either by adding more phases or by increasing the precision of predictions. The rest of their paper is concerned with increasing the precision of predictions, but the issue of power enhancement by increasing the number of phases was taken up by Ferron and Onghena (1996). They considered six-phase designs with random assignment of treatments to phases of four to eight observations in length. As in a randomized treatment design with six observation times—rather than six phases—as the experimental units, the number of possible data arrangements would be 20 (i.e., minimum possible a value=0.05). However, in the phase design this is not a good estimate of the relative power of the associated randomization test (i.e., for a given effect size and α level for rejection of the null hypothesis) because a phase mean will be a more precise measure than a single observation. In short, we can expect power to be beneficially affected by the increase in measurement precision provided by increases in phase length.

Onghena (1994) also considered the effect of various levels of phase length and autocorrelation on power for a range of effect sizes with this design. The results are fairly complex but it is clear that longer phase lengths and positive autocorrelations are associated with higher power, and that for large effect sizes at least, power is adequate. Certainly, power is far more satisfactory than in an ABAB design investigated by Ferron and Ware (1995), which had three randomly assigned interventions and about the same total number of observations, even though that design had more than six times as many possible arrangements.

Conclusions

What general advice can we offer in the light of these early findings? It would probably be sensible to avoid a straight AB design unless a very large effect is anticipated. Little is known as yet about the power of extensions of that design, particularly when autocorrelation is present, but it seems likely that these will be capable of delivering substantial improvements. Phase designs with enough phases to be able to achieve an α=0.05 when there is random allocation of treatments to phases (minimum number of phases=6) are well worth considering when the logic of the experiment makes it possible to vary the order of phases. Randomized treatment designs, with assignment of treatments to individual observation times, are attractive from the point of view of power, although they may be impractical for the investigation of many research questions. It must be conceded that the systematic exploration of power considerations for single-case designs has barely begun, but it seems likely that the usefulness of some popular phase designs may need to be reassessed as power information accumulates. We predict that the search will be on for extensions and modifications of standard designs that yield higher power estimates. This, we believe, would be good news. The neglect of power considerations in single-case experiments has probably resulted in much well-intentioned but wasteful effort. In the meantime, we would do well to focus on those aspects of experimental design, such as control and reliability, that we know can improve the statistical power of our tests, even if we cannot always quantify the improvement.

Chapter 12
Creating Your Own Randomization Tests

The range of potential randomization tests is limited only by the ingenuity of researchers in designing new randomization procedures. In principle, a valid randomization test can be developed for application to data from any design containing an element of random assignment. Of course, someone has to develop the test. Fortunately, there is probably a relatively small number of core randomization designs, each with possibilities for extension and modification. We have tried to provide a representative sample of designs, along with Minitab, Excel, and SPSS macros, for carrying out randomization tests on data derived from them. In this final chapter we aim to help researchers who are interested in writing macros to match their own random assignment procedures or others that they see described in the literature. We mentioned in chapters 4 and 10 that we decided to use random sampling algorithms for all of our macros, rather than exhaustive generation of all possible data arrangements, to maintain consistency and to permit trade-off between time costs and power. Another gain is that algorithms based on exhaustive data arrangements tend to be highly specific, whereas those based on random sampling from the possible arrangements tend to be somewhat more general and therefore less in need of modification. This is important because modification of existing macros turns out to be problematic for all but the most trivial changes, such as number of arrangements to be randomly sampled.

When we began this project, we envisaged a final chapter providing guidance on how to customize our macros to deal with design modifications. As we progressed, it became increasingly clear that this was not going to work. Unfortunately, quite small modifications to designs can necessitate very substantial modifications to an existing macro. Almost invariably, it turns out to be simpler to start from scratch, using existing macros as a source of ideas for developing the new one. For example, at a superficial level, it looks as though our Design 2 (ABA) should be a rather minor modification of our Design 1 (AB). In fact, adding a reversal phase meant that an extra level of complexity was introduced because the available withdrawal points depend on the point selected for intervention. Certainly, insights gained from writing code for the AB design were useful when working on the macro for the ABA design, but there was not a lot of scope for cutting and pasting sections of code. To have attempted to do so would have increased the complexity of the task rather than simplifying it. This was a very general observation. Almost every apparently slight change in design that we have looked at turned out to introduce a new problem that was best solved by taking what had been learned from solving the "parent" macro without being too attached to the particulars of that solution. Part of the problem is that it is not possible to use the standard functions to do things like ANOVA and ANCOVA because the output cannot be stored (or not in a place where you can use it), so the test statistics have to be constructed for each design.

We had hoped to be able to demonstrate customization of macros to deal with a range of design modifications in the literature, such as the restriction of allowable randomizations in the single-case randomized treatment design (Onghena & Edgington, 1994) and removal of trend by using trial number as a covariate in an AB design (Edgington, 1995). For the

reasons outlined earlier, we abandoned that goal. Of course, we could have provided new macros for modifications such as these, but the list of possible design modifications is virtually endless, and we could not provide macros for them all. Instead, we offer some suggestions for how interested readers might go about writing a new macro for themselves.

STEPS IN CREATING A MACRO FOR A RANDOMIZATION TEST

Anyone wanting to write a macro for a randomization test in the packages we used, or in other statistical packages, can break the problem into the following steps:

1. Work out how to calculate the required test statistic for the actual data.
2. Find a way to do the rearrangements of the data.
3. Apply the result of Step 1 to each arrangement and store the arrangement test statistic.
4. Compare the arrangement test statistics with the actual test statistic and count those that are at least as extreme.
5. Calculate the probability.

Of these steps, Step 2 is likely to be the most difficult and may require considerable ingenuity. Steps 4 and 5 need to be modified and repeated in cases where both one- and two-tailed tests are possible.

To complete Step 1, it is usually possible to use the menus and track the equivalent commands, which can then be used to help in writing the macro. Minitab puts the commands in the session window whenever menus are used to get results, so these are easy to copy. Excel has a "record macro" facility available using the menu route *Tools>Macro>Record New Macro*. Using this it is possible to view the Visual Basic commands used to achieve a result via the menus. Of course it is not necessary to have a whole macro to record; you can just do a small part of it and view the Visual Basic commands. SPSS also allows you to view the equivalent statements in the command language when you use a menu route. There are two ways to achieve this. One way is to select *Paste* instead of *OK* in the dialog box when the menu selections have been completed. This will result in the commands being displayed in a *Syntax Editor* window instead of being executed. The commands can then be executed by selecting *Run* from the Syntax Editor menu. An alternative method makes the commands appear in the *Output Viewer* window along with the results. To do this, before executing the commands, use the menu route *Edit>Options* and click the *Viewer* tab, then click the *Display commands in log* check box, and click OK. However, matrix language commands are available only in the Syntax Editor, so this device cannot be used to enable you to find appropriate matrix commands.

The online help in all three of these packages is useful, but sometimes you need a manual as well. The designs covered in this book provide a variety of methods of dealing with the problems. You should, however, be aware that we have sometimes sacrificed possible simplifications in the interest of greater generality. Nonetheless, the annotated macros should provide the user with a useful crib when attempting to write a new one.

WRITING YOUR OWN MACROS

Users who want to modify any of our designs should study the relevant annotated macros in the chosen package and see what can be borrowed. For example, if you want to try an ABAB design, consider the macros for AB and ABA to see how the new level of complexity is handled. In this case the problem is that the available withdrawal points depend on the chosen intervention point, so you cannot just repeat the same process when choosing the withdrawal point. You must choose the intervention point, then conditional on that, choose the withdrawal point. To move to ABAB you will have to make the available points for the second intervention conditional on the first intervention and the withdrawal. To do an ABAB design with several participants, the process of choosing intervention and withdrawal points has to be repeated for each participant: Look at Designs 3 and 4 to see how this is handled.

Design 5 is so simple that unequal-sized treatment groups are allowed. To allow this in Design 6, with treatments applied within blocks, would add considerably to the difficulty of managing the rearrangements, because rearrangements have to be done within blocks. On the other hand, it would not be very hard to introduce a modification to allow less orderly data entry: You would just have to sort the data into blocks before starting the main work of the macro. All of the packages allow sorting, and although we have written the macros for Designs 6 through 8 on the assumption that the data are entered in a very orderly way into the worksheet, the order could be imposed within the macros by making use of the sort facilities. In fact you may have noticed that this facility has to be used in Design 7 (a 2×2 factorial design) to deal with the second factor anyway.

Design 7 has equal numbers of observations at each combination of factor levels. Unequal numbers would introduce the same difficulty as with Design 6. An extra factor, also at two levels, could be accommodated by performing the rearrangements for Factor 1 within combinations of levels of Factors 2 and 3, and similarly for the other factors. However, to introduce extra levels of a factor you would have to use a different test statistic, perhaps the RSS as in Designs 5 and 6. Because you need to store your test statistics in a way that permits subsequent comparison with the actual test statistic, you may not be able to use the easiest way your package offers for calculating a statistic such as the RSS. Differences between two means are usually fairly easy to compute (although SPSS users may notice that it is not completely trivial in the matrix language), but the RSS takes more work. In some cases the treatment SS is an easier alternative and is equivalent.

We hope that our designs and the notes about them will give other people good ideas about how to proceed in new cases, and perhaps in other packages. We decided to work with high-level languages because we believe that people are more likely to try something new to them in a familiar computing environment. However, there is a cost: A lower level language than we have used would give more scope for complexity, both because of efficiency (programs will run in less time) and because of the greater flexibility of lower level languages. For example, the work of Edgington (1995) in FORTRAN goes well beyond what we have done here, and anyone wishing to attempt a design for which ours give little help might turn to Edgington's FORTRAN programs for ideas.

We have concluded that, in general, customization of our macros is not a realistic option. There are, however, some procedural modifications to our designs that can be accomplished without the need to write new code, and we finish with examples of these.

TINKERING: CHANGING VALUES WITHIN A MACRO

There is one kind of modification that is straightforward. Wherever limits of some kind have been set, they can be altered by editing the macro. We explained in chapters 6 through 8 how to do this for each package with respect to the number of data arrangements to be randomly sampled and, in the case of SPSS (chap. 8), the number of permitted loops.

DESIGN MODIFICATIONS WITHOUT MACRO CHANGES

Some design modifications are possible without modifying an existing macro. These will be modifications that do not alter the randomization procedure. A good example of this, which we discussed briefly in chapter 2, would be when a researcher wished to retain an element of response-guided methodology to ensure baseline stability, without compromising the internal validity of an AB phase experiment. In this modification (Edgington, 1975), the researcher continues with preexperimental baseline observations until a stability criterion is reached—the response-guided part of the design. Then a treatment intervention point is randomly determined within a predetermined number of observations in the experimental part of the design. Only the preintervention (baseline) and postintervention (treatment) data from the experimental part of the design are analyzed, and this can be done using the randomization test provided for Design 1. Similarly, it would be acceptable to prolong the treatment phase beyond the point specified in a randomization design to maximize a therapeutic effect (Gorman & Allison, 1996), provided that only the data specified by the experimental design were subjected to the randomization test.

DATA MODIFICATION PRIOR TO ANALYSIS

There may be circumstances in which some preliminary work on the data can result in something appropriate for one of our designs. We mentioned earlier that Edgington (1995) proposed a randomization test in which trend in an AB design was removed by using trial number as a covariate. Although there is no way of achieving the necessary modification by cutting and pasting chunks of code in our Design 1 macros—major rewriting of code would be called for—a solution based on preanalysis work on the data may be possible. For example, if data from an AB design show a strong trend superimposed on what may be a change in level, a regression line could be fitted to the baseline data to remove this trend from all of the data before using a Design 1 macro. In his discussion of the theory of randomization tests, Edgington (1995) explicitly rejected the solution of using residuals derived from a baseline regression equation. He argued that validity would not be assured because the principle of a closed reference set of data arrangements is violated. By this, he meant that the size (and therefore the rank) of the statistic for any particular data arrangement would differ according to which intervention point was randomly selected, because the residuals would be based on different regression equations. We accept this argument

and agree that the "correct" solution would be to compute the ANCOVA statistic for each data arrangement. However, if no "off the shelf" covariance solution was available, or even if it was available, but only in an unfamiliar programming language (e.g., Fortran) in an unfamiliar computing environment (e.g., DOS), someone used to using only standard statistical packages might well abandon their intention to carry out any randomization test at all. Provided that the *p* value obtained is treated as an approximate guide to the correct value, it may sometimes be preferable to performing no statistical analysis at all (see our discussion of the general issue in chap. 10). In time, no doubt, empirical studies of the robustness of randomization tests when assumptions such as "closure of reference sets" are not met will provide some guidance, as they currently do with regard to violations of assumptions required for parametric tests. In the meantime, any reader in doubt about the suitability of our regression solutions for their purposes would do well to consult Edgington's (1995) theoretical chapter before making a decision. On the assumption that some readers may decide that our approach could provide some help in deciding whether a variable is worth pursuing, we now provide two specific examples to illustrate how the regression residual transformation of data for analysis using our Design 1 might work.

Downward Slope During Baseline

We begin with an example provided by Edgington (1995). In his example, a downward trend in the data (maybe a fatigue or boredom effect) may make it difficult to infer a treatment effect, even though introduction of the treatment results in a sudden, marked elevation of the observation score. He showed that even a very dramatic effect of the treatment could result in there being no difference between baseline and treatment means. He made the point with the following very orderly and extreme data set:

baseline: 9 8 7 6 5 4 3 treatment: 9 8 7 6 5 4 3

His solution was to remove the trend by using trial number as a covariate. An alternative solution that makes it possible to use our macros for Design 1 is to fit a regression line to the baseline data and then use the residuals in place of the raw data for analysis with a Design 1 macro. If we fit a regression line to the first seven points (the baseline data), it will of course be a perfect fit with these artificial data and the baseline residuals (i.e., the deviations of the baseline scores from the regression line) will all be zero.

Regression Worksheet Box	
data	**obs. no.**
9	1
8	2
6	4
5	5
4	6
3	7

It is easy with the made-up data, but the reader may wish to know how the residuals would be calculated with real data. Any statistical package will provide a means of obtaining a regression equation (an equation that defines the best fitting straight line through the points in terms of the slope of the line and its intercept on the vertical or score axis). For example, in Minitab, Excel, or SPSS, the data would be entered in one column and the observation number in another, as shown in the Regression Worksheet Box. Then the menu route within Minitab would be *Statistics>Regression>Regression,* with data entered in the *Response* box and obs. no. in the *Predictors* box. The equivalent menu route in SPSS would be *Analyze > Regression>Linear,* with data entered in the *Dependent* box and obs. no. in the *Independent(s)* box. In Excel, it is possible to enter the formulae=*index(linest (a2:a8, b2:b8), 1)* and =*index (linest (a2:a8, b2:b8), 2),* to obtain the slope and intercept, respectively. It is also simple in any of the packages to get the residuals listed or stored in a new column. In this example, the regression equation is: *data=10−(1×obs. no.)*.

Residuals Worksheet Box

limits	data	phase
14	0	0
3	0	0
3	0	0
	0	0
	0	0
	0	0
	7	1
	7	1
	7	1
	7	1
	7	1
	7	1

In SPSS the intercept and slope values are labeled Unstandardized Coefficients (i.e., B) for the Constant and obs. no., respectively. The equation tells us that the intercept is at a score of 10 and for every increment of one for obs no., the data score reduces by one. This can be clearly seen in Fig. 12.1. As we said earlier, and as is obvious in Fig. 12.1, the residuals for the baseline data are all zero.

Now we need to subtract the regression from the data points in the treatment phase (i.e., data −10+obs. no.). For example, for the first data point in the treatment phase, the difference is 9−10+8=7. In fact, as can be seen in Fig. 12.1, all of the treatment data points are 7 score units above the baseline regression line. Therefore, the complete set of transformed data comprises seven zeros preintervention and seven 7s postintervention. If we assume that the design specified at least three observations in each phase and that the intervention at observation Period 8 was randomly chosen, we can use Design 1 to analyze the data. The worksheet entries are shown in the Residuals Worksheet Box. There will be nine possible intervention points, so the smallest possible p value when the difference between means is computed for all possible data splits is 1/9=0.11. As the difference between means is clearly greatest for the actual intervention point, a p value of 0.11, within random sampling error, is obtained when one of the Design 1 macros is run on the transformed data.

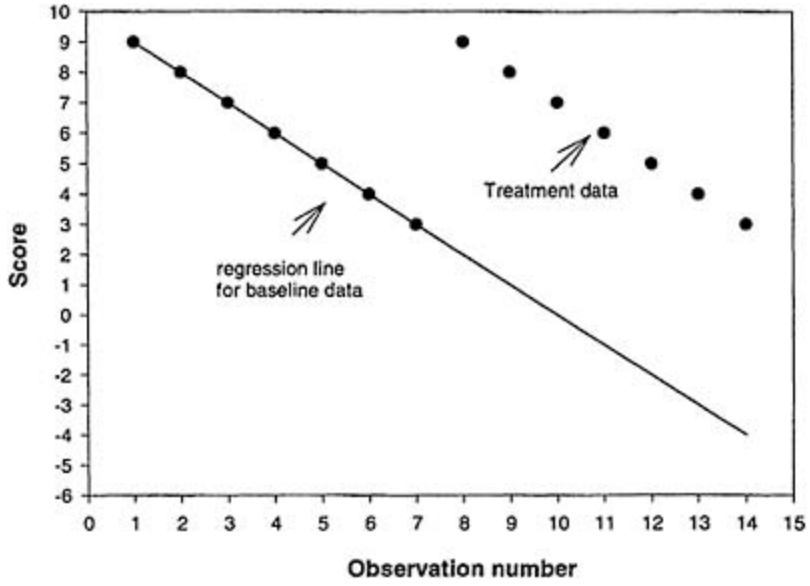

FIG. 12.1. Idealized data showing a treatment effect combined with a downward trend in the data.

Upward Slope During Baseline

An upward trend in the data (maybe a practice effect) may make it difficult to infer a treatment effect, even though the scores appear to increase after the intervention point. Consider the following made-up data:

baseline: 1 2 3 4 5 6 7 treatment: 10 11 12 13 14 15 16

With a clear upward baseline trend such as this, there is no point in testing the difference between pre- and postintervention scores; finding that postintervention scores are higher would not allow us to infer that the treatment had an effect, as we would expect them to be higher anyway due to continuation of the trend. As in the previous example, we can obtain the regression equation for baseline data and then use residuals around that regression line in place of the raw data for both baseline and treatment phases. If there really was an immediate effect of the treatment, the sum of residuals in the baseline phase would be zero (i.e., equally distributed above and below the regression line) and the sum of the residuals in the treatment phase would be positive (i.e., mostly above the regression line). The randomization test would give the probability (under the null hypothesis) of a difference between baseline and treatment residuals as extreme as that actually found.

In this example, the regression equation is *data=0+(1×obs. no.)*. The equation tells us that the intercept is at zero and for every increment of one for obs no., the data score increases by one. This can be clearly seen in Fig. 12.2, where it is also obvious that the residuals for the baseline data are all zero. As before, we now need to subtract the regression from the data points in the treatment phase (i.e., data −0− obs. no.). The residual for

the first data point in the treatment phase is 10–0−8=2, and it is obvious from Fig. 12.2 that all of the treatment data points are 2 score units above the baseline regression line.

The transformed data set now comprises seven zeros (for the baseline phase) followed by seven 2s (for the treatment phase). If we assume, once again, that the design specified at least three observations in each phase and that the intervention at observation Period 8 was randomly chosen, again we can use Design 1 to analyze the data. The worksheet entries would be as shown in the Residuals Worksheet Box, except that the seven 7s would be replaced by seven 2s. As before, there will be nine possible intervention points, so the smallest possible *p* value when the difference between means is computed for all possible data splits is 1/9=0.11. Once again, as the difference between means is clearly greatest for the actual intervention point, a *p* value of 0.11, within random sampling error, is obtained when one of the Design 1 macros is run on the transformed data.

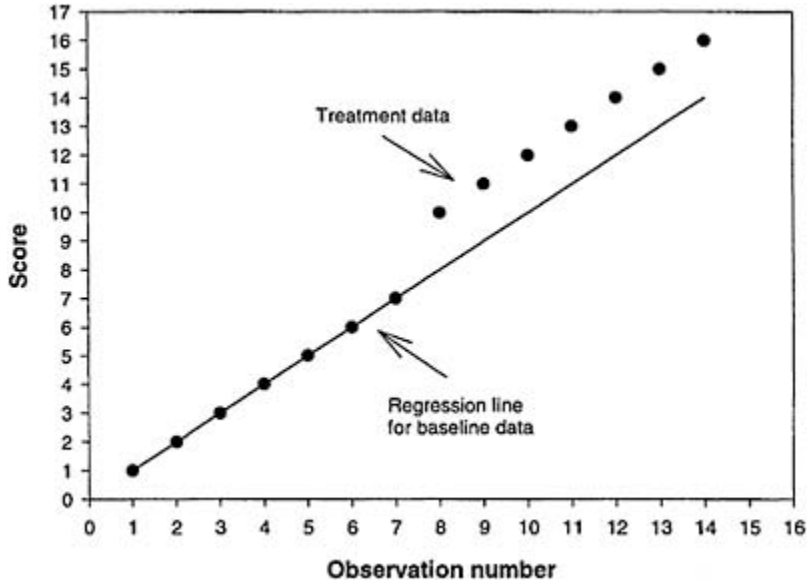

FIG. 12.2. Idealized data showing a treatment effect combined with an upward trend in the data.

We finish with a more realistic set of data for the upward trend example. Suppose that 30 observations were specified, with at least 4 in each phase, that the randomly selected intervention point was observation Period 22, and the obtained scores were as follows:

baseline: 1 3 2 4 6 5 8 7 9 10 12 13 13 15 15 15 17 20 18 19 21
treatment: 25 26 26 28 262931 32 32

The data, together with the baseline regression line, are shown in Fig. 12.3. In this example, the regression equation is *data=0.0195+(0.992× obs. no.)*. As in the previous example, the sum of the baseline residuals is necessarily zero and the difference between individual

baseline and treatment data points and the regression line (i.e., baseline and treatment residuals) is given by *data−0.0195−(0.991×obs. no.)*. The baseline residuals can be saved in a worksheet when the regression analysis is run in Minitab or SPSS and the treatment residuals can be generated in a worksheet by using the *Calc>Calculator* menu route in Minitab or the *Transform>Compute* route in SPSS. In Excel, formulae can be entered to obtain both baseline and treatment residuals. The residual for the first data point in the treatment phase is 25–0.0195−(0.991×22)=3.1785. The entries in the worksheet would be:

- Limits column—In the top three rows: 30, 4, 4
- Data column—Baseline residuals (−0.19, 0.82, −1.17, −0.16, 0.85, −1.14, 0.87, −1.12, −0.11, −0.10, 0.90, 0.91, −0.08, 0.93, −0.06, −1.05, −0.04, 1.97, −1.02, −1.01, 0.00) in the first 21 rows, followed by the treatment residuals (3.18, 3.19, 2.20, 3.21, 0.21, 2.22, 3.23, 3.24, 2.25).
- Phase column—21 zeros (baseline), followed by 9 ones (treatment).

The number of possible intervention points was 23, so the lowest possible *p* value with all possible data splits was 1/23=0.043. When one of the Design 1 macros was run, the obtained test statistic (difference between residual means) for the actual data was 2.55 and the *p* value (one-tailed and two-tailed) was 0.048. Therefore, the difference between baseline and treatment residuals around the baseline just reached statistical significance at the 5% level.

There are two useful lessons to be taken from this example. First, with a *p* value so close to the critical value, it is quite likely that on some runs of the macro the *p* value would fall on the "wrong side" of the cr iti cal value. A possible strategy in this situation would be to

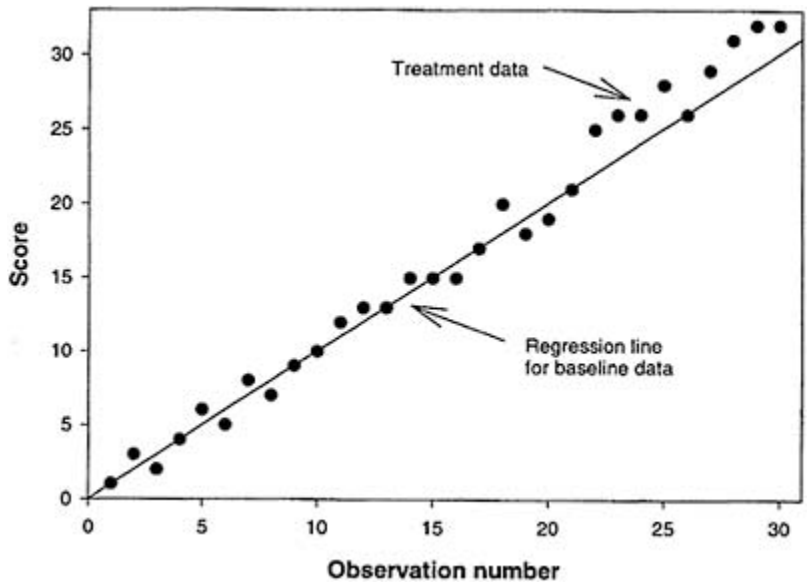

FIG. 12.3. More realistic data showing a possible treatment effect combined with an upward trend in the data.

boost confidence in the statistical decision by increasing the number of random samples of data splits. When we increased the number of samples from 2000 to 10000 for this data set, we obtained a *p* value of 0.041, which might make us rather more comfortable about reporting a significant effect. On the other hand, it might be argued (E.S.Edgington, personal communication, April 4,2000) that there is a need for adjustment of the *p* value when sequential tests of this kind are carried out. Certainly, it seems safer to make a judgment about what constitutes a sufficiently large sample of random rearrangements before conducting the test. The second thing to note about this example is that if, following the score jump immediately after the treatment was introduced, scores had continued to increase at a greater rate than in the baseline, the *p* value would have decreased. This is because data splits closer to the end of the treatment observations would produce bigger differences than the difference at the point of intervention. This highlights the importance of carefully considering precisely what the hypothesis under test is supposed to be. This data transformation was designed to test the hypothesis that introduction of the treatment would have an immediate "one-off" effect. If the hypothesis had been that the treatment would have a cumulative effect over trials, the data transformation could not have been used in a test of that hypothesis.

SOURCES OF DESIGN VARIANTS AND ASSOCIATED RANDOMIZATION TESTS

Edgington's (1995) book is a rich source of ideas for variants of the basic designs, and we found papers by Chen and Dunlap (1993), Ferron and Ware (1994), Hayes (1998), Onghena (1992), Onghena and Edgington (1994), Onghena and May (1995), and Wampold and Furlong (1981) particularly helpful. So far as the provision of randomization tests for single-case designs is concerned, the SCRT package makes it possible to implement randomization tests for a fairly wide range of design variants, including restricted alternating treatments designs and extensions of the ABA design to include additional phases. We hope that researchers who are tempted into trying out randomization tests that we have provided within a familiar package will be sufficiently encouraged to go on to explore the SCRT package.

References

Allison, D.B., Silverstein, J.M., & Gorman, B.S. (1996). Power, sample size estimation, and early stopping rules. In R.D.Franklin, D.B.Allison, & B.S. Gorman (Eds.), *Design and analysis of single-case research* (pp. 335–371). Mahwah, NJ: Lawrence Erlbaum Associates.

Baker, R.D. (1995). Modern permutation test software. In E.S.Edgington, *Randomization tests* (3rd ed., pp. 391–401). New York: Dekker.

Barlow, M.D.H., & Hersen, M. (1984). *Single case experimental designs: Strategies for studying behavior change* (2nd ed., pp. 285–324). New York: Pergamon.

Borenstein, M., Rothstein, H., & Cohen, J. (1997). SamplePower™ 1.0. Chicago: SPSS Inc.

Box, G.E.P., & Jenkins, G.M. (1976). *Time series analysis, forecasting and control.* San Francisco: Holden-Day.

Bradley, J.V. (1968). *Distribution-free statistical tests.* Englewood Cliffs, NJ: Prentice Hall.

Bryman, A., & Cramer, D. (1990). *Quantitative data analysis for social scientists.* London: Routledge.

Busk, P.L., & Marascuilo, L.A. (1992). Statistical analysis in single-case research: Issues, procedures, and recommendations, with applications to multiple behaviors. In T.R.Kratochwill & J.R.Levin (Eds.), *Single-case research design and analysis: New directions for psychology and education* (pp. 159–185). Hillsdale, NJ: Lawrence Erlbaum Associates.

Campbell, D.T., & Stanley, J.C. (1966). *Experimental and quasi-experimental designs for research.* Chicago: Rand McNally.

Chen, R.S., & Dunlap, W.P. (1993). SAS procedures for approximate randomization tests. *Behavior Research Methods, Instruments, & Computers, 25,* 406–409.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale, NJ: Lawrence Erlbaum Associates.

Dugard, P., & Todman, J. (1995). Analysis of pretest-posttest control group designs. *Educational Psychology, 15,* 181–198.

Edgington, E.S. (1969). Approximate randomization tests. *The Journal of Psychology, 72,* 143–149.

Edgington, E.S. (1975). Randomization tests for one-subject operant experiments. *The Journal of Psychology, 90,* 57–68.

Edgington, E.S. (1984). Statistics and single case analysis. In M.Hersen, R.M. Eisler, & P.M.Miller (Eds.), *Progress in behavior modification* (Vol. 16, pp. 83–119) Orlando, FL: Academic.

Edgington, E.S. (1987). *Randomization tests* (2nd ed.). New York: Dekker.

Edgington, E.S. (1992). Nonparametric tests for single-case experiments. InT. R.Kratochwill & J.R.Levin (Eds.), *Single-case research design and analysis* (pp. 133–157). Hillsdale, NJ: Lawrence Erlbaum Associates.

Edgington, E.S. (1995). *Randomization tests* (3rd ed.). New York: Dekker.

Edgington, E.S. (1996). Randomized single-subject experimental designs. *Behaviour Research and Therapy, 34,* 567–574.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans.* Philadelphia: Society for Industrial and Applied Mathematics.

Ferron, J., & Onghena, P. (1996). The power of randomization tests for single-case phase designs. *The Journal of Experimental Education, 64,* 231–239.

Ferron, J., & Ware, W. (1994). Using randomization tests with responsive singlecase designs. *Behaviour Research and Therapy, 32,* 787–791.

Ferron, J., & Ware, W. (1995). Analyzing single-case data: The power of randomization tests. *The Journal of Experimental Education, 63,* 167–178.

Fisher, R.A. (1935). *The design of experiments.* Edinburgh, Scotland: Oliver & Boyd.

Franklin, R.D., Allison, D.B., & Gorman, B.S. (1996). Introduction. In R.D. Franklin, D.B.Allison, & B.S.Gorman (Eds.), *Design and analysis of single-case research* (pp. 1–11). Mahwah, NJ: Lawrence Erlbaum Associates.

Franklin, R.D., Gorman, B.S., Beasley, T.M., & Allison, D.B. (1996). Graphical display and visual analysis. In R.D.Franklin, D.B.Allison, & B.S.Gorman (Eds.), *Design and analysis of single-case research* (pp. 119–158). Mahwah, NJ: Lawrence Erlbaum Associates.

Gentile, J.R., Roden, A.H., & Klein, R.D. (1972). An analysis of variance model for the intrasubject replication design. *Journal of Applied Behavior Analysis, 5,* 193–198.

Good, P. (1994). *Permutation tests: A practical guide to resampling methods for testing hypotheses.* New York: Springer-Verlag.

Gorman, B.S., & Allison, D.B. (1996). Statistical alternatives for single-case designs. In R.D.Franklin, D.B.Allison, & B.S.Gorman (Eds.), *Design and analysis of single-case research* (pp. 159–214). Mahwah, NJ: Lawrence Erlbaum Associates.

Hayes, A.F. (1998). SPSS procedures for approximate randomization tests. *Behavior Research Methods, Instruments, & Computers, 30, 536–543.*

Howell, D.C. (1997). *Statistical methods for psychology* (4th ed.). Belmont, CA: Duxbury.

Jeffreys, H. (1939). *Theory of probability.* Oxford, UK: Clarendon.

Kazdin, A.E. (1973). Methodological and assessment considerations in evaluating reinforcement programs in applied settings. *Journal of Applied Behavior Analysis, 6,* 517–531.

Kazdin, A.E. (1975). *Behavior modification in applied settings.* Homewood, IL: Dorsey.

Kazdin, A.E. (1976). Statistical analyses for single-case experimental designs. In M.Hersen & D.H.Barlow (Eds.), *Single case experimental designs: Strategies for studying behavior change* (pp. 265–316). New York: Pergamon.

Kazdin, A.E. (1980). Obstacles in using randomization tests in single-case experimentation. *Journal of Educational Statistics, 5, 253–260.*

Kazdin, A.E. (1982). *Single-case research designs: Methods for clinical and applied settings.* London: Oxford University Press.

Kazdin, A.E. (1984). Statistical analyses for single-case experimental designs. In D.H.Barlow & M.Hersen (Eds.), *Single case experimental designs: Strategies for studying behavior change* (2nd ed., pp. 285–324). New York: Pergamon.

Levin, J.R., Marascuilo, L.A., & Hubert, L.J. (1978). N=nonparametric randomization tests. In T.S.Kratochwill (Ed.), *Single subject research: Strategies for evaluating change* (pp. 167–196). New York: Academic.

Lindley, D.V. (1965). *Introduction to probability and statistics from a Bayesian viewpoint.* Cambridge, UK: Cambridge University Press.

Manly, B.F.J. (1991). *Randomization and Monte Carlo methods in biology.* London: Chapman & Hall.

Marascuilo, L.A., & Busk, P.L. (1988). Combining statistics for multiple-baseline AB and replicated ABAB designs across subjects. *Behavioral Assessment, 10,* 1–28.

Matyas, T.A., & Greenwood, K.M. (1996). Serial dependency in single-case time series. In R.D.Franklin, D.B.Allison, & B.S.Gorman (Eds.), *Design and analysis of single-case research* (pp. 215–243). Mahwah, NJ: Lawrence Erlbaum Associates.

May, R.B., Masson, M.E.J., & Hunter, M.A. (1990). *Application of statistics in behavioral research.* New York: Harper & Row.

Minium, E.W., King, B.M., & Bear, G. (1993). *Statistical reasoning in psychology and education* (3rd ed.). New York: Wiley.

Neyman, J., & Pearson, E.S. (1967). *Collected joint statistical papers.* Cambridge, UK: Cambridge University Press.

Noreen, E.W. (1989). *Computer-intensive methods for testing hypotheses: An introduction.* New York: Wiley

Onghena, P. (1992). Randomization tests for extensions and variations of ABAB single-case experimental designs: A rejoinder. *Behavioral Assessment, 14,* 153–171.

Onghena, P. (1994). *The power of randomization tests for single-case designs.* Unpublished doctoral dissertation, Katholieke Universiteit Leuven, Belgium.

Onghena, P., & Edgington, E.S. (1994). Randomized tests for restricted alternating treatments designs. *Behaviour Research and Therapy,* 32, 783–786.

Onghena, P., & May, R.B. (1995). Pitfalls in computing and interpreting randomization *p* values: A commentary on Chen and Dunlap. *Behavior Research Methods, Instruments, & Computers,* 27, 408–411.

Onghena, P., & Van Damme, G. (1994). SCRT1.1: Single case randomization tests. *Behavior Research Methods, Instruments, & Computers, 26, 369.*

Onghena, P., & Van Den Noortgate, W. (1997). Statistical software for microcomputers: Review of StatXact-3 for Windows and LogXact (version 2) for Windows. *British Journal of Mathematical and Statistical Psychology, 50, 370–373.*

Parsonson, B.S., & Baer, D.M. (1978). The analysis and presentation of graphic data. In T.R.Kratochwill (Ed.), *Single subject research: Strategies for evaluating change* (pp. 101–165). New York: Academic.

Pitman, E.J.G. (1937). Significance tests which maybe applied to samples from any populations. *Journal of the Royal Statistical Society: Section B, 4,* 119–130.

Primavera, L.H., Allison, D.B., & Alfonso, V.C. (1996). Measurement of dependent variables. In R.D.Franklin, D.B.Allison, & B.S.Gorman (Eds.), *Design and analysis of single-case research* (pp. 41–91). Mahwah, NJ: Lawrence Erlbaum Associates.

Remington, R. (1990). Methodological challenges in applying single case designs to problems in AAC. In J.Brodin & E.Bjorck-Akesson (Eds.), *Proceedings from the first ISAAC Research Symposium in Augmentative and Alternative Communication* (pp. 74–78). Stockholm: Swedish Handicap Institute.

Shaughnessy, J.J., & Zechmeister, E.B. (1994). *Research methods in psychology* (3rd ed.). New York: McGraw-Hill.

Sidman, M. (1960). *Tactics of scientific research: Evaluating experimental data in psychological research.* New York: Basic Books.

Siegel, S., & Castellan, N.J., Jr. (1988). *Nonparametric statistics for the behavioral sciences* (2nd ed.). Singapore: McGraw-Hill.

Skinner, B.F. (1956). A case history in scientific method. *American Psychologist*, *11,* 221–233.

Skinner, B.F. (1966). Operant behavior. In W.K.Honig (Ed.), *Operant behavior: Areas of research and application* (pp. 12–32). New York: Appleton-Century-Crofts.

Todman, J., & Dugard, P. (1999). Accessible randomization tests for single-case and small-*n* experimental designs in AAC research. *Augmentative and Alternative Communication, 15,* 69–82.

Wampold, B.E., & Furlong, M.J. (1981). Randomization tests in single-subject designs: Illustrative examples. *Journal of Behavioral Assessment, 3,* 329–341.

Wilcox, R.R. (1987). *New statistical procedures for the social sciences.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Author Index

Subject Index