



S+SEQTRIAL 2

User's Manual

December 2002

Insightful Corporation
Seattle, Washington

**Proprietary
Notice**

Insightful Corporation owns both this software program and its documentation. Both the program and documentation are copyrighted with all rights reserved by Insightful Corporation.

The correct bibliographical reference for this document is as follows:

S+SEQTRIAL 2 User's Manual, Insightful Corporation, Seattle, WA.

Printed in the United States.

Copyright Notice Copyright © 2002, Insightful Corporation. All rights reserved.

Insightful Corporation
1700 Westlake Avenue N, Suite 500
Seattle, WA 98109-3044
USA

Trademarks

S-PLUS is a registered trademark, and StatServer, S-PLUS Analytic Server, S+SDK, S+SPATIALSTATS, S+FINMETRICS, and S+WAVELETS are trademarks of Insightful Corporation; S and New S are trademarks of Lucent Technologies, Inc.; Intel is a registered trademark, and Pentium a trademark, of Intel Corporation; Microsoft, Windows, MS-DOS, and Excel are registered trademarks, and Windows NT is a trademark of Microsoft Corporation. Other brand and product names referred to are trademarks or registered trademarks of their respective owners.

ACKNOWLEDGMENTS

The Research and Development of the S+SEQTRIAL module was partially supported by National Institutes of Health Small Business Innovative Research (NIH SBIR) Grant 2R44CA69992-02 awarded to Andrew Bruce (Principal Investigator) of Insightful Corp. S+SEQTRIAL was jointly developed by Scott Emerson of the University of Washington and Andrew Bruce, Edward Chao, Steve Allan, Charles Roosen, Tim Hesterberg, and Sylvia Isler of Insightful Corp. This manual was written by Scott Emerson, Tim Hesterberg, and Len Kannapell.

CONTENTS

Chapter 1	Introduction	1
Overview		2
S+SEQTRIAL Features		8
Using This Manual		11
Technical Support		15
Chapter 2	Getting Started	17
Overview		18
Using S+SEQTRIAL		19
S+SEQTRIAL Graphical User Interface		22
Typical Workflow		27
Chapter 3	Tutorial	29
Overview		30
Launching the S+SEQTRIAL Dialog		31
Designing a Fixed Sample Test		33
Designing a Group Sequential Test		40
Advanced Group Sequential Designs		53
Where to Go From Here		56
Chapter 4	Specifying a Fixed Sample Design	57
Overview		58
Clinical Trial Design in S+SEQTRIAL		59

Types of Clinical Trials	64
Types of Probability Models	68
Creating a New Design	73
Updating an Existing Design	78
Design Parameters	80
Chapter 5 Specifying a Group Sequential Design	99
Overview	101
Group Sequential Design Concepts	102
Boundary Scales	112
Boundary Shape	118
Issues in Choosing a Stopping Rule	130
Basic Group Sequential Designs	132
Advanced Group Sequential Designs	135
Boundary Scale Parameters	140
Constrained Boundaries	158
Chapter 6 Evaluating Designs	161
Overview	162
Stopping Boundaries	164
Inference on the Boundaries	175
Sample Size Distributions	176
Power Functions	180
Stopping Probabilities	185
The Results Tab	187
The Plot Tab	195
Chapter 7 Monitoring a Trial	201
Overview	202
Reasons for Monitoring a Clinical Trial	203

Statistical Issues in Monitoring a Clinical Trial	204
Computing Modified Boundaries	206
Specifying a Monitoring Procedure	223
Evaluating A Monitoring Rule	227
Chapter 8 Reporting Results	229
Overview	230
Adjusted Point Estimators	231
Adjusted Confidence Intervals	233
Adjusted P-Values	235
Obtaining Adjusted Estimates and P-Values	236
Chapter 9 Case Study #1: Hybrid Tests	239
Overview	240
Limitations of the Standard Designs	242
Hybrid Tests	253
Chapter 10 Case Study #2: Unplanned Analyses	261
Overview	262
Designing a Fixed Sample Test	263
Designing a Group Sequential Test	267
Conducting the Clinical Trial	272
Performing a Sensitivity Analysis	283
Chapter 11 Case Study #3: Unexpected Toxicities	289
Overview	290
Designing a Fixed Sample Test	291
Designing Group Sequential Tests	295
Data-Imposed Stopping Rules	306

Contents

Chapter 12 Case Study #4: Efficiency Concerns	309
Overview	310
Initial Protocol	311
Sponsor Safety Boundary	316
DSMB Safety Boundary	323
DSMB Efficiency Boundary	329
Sponsor Futility Boundaries	335
Advanced Boundary Scales	341
Appendix A: Functions by Category	345
Appendix B: Function Reference	351
Bibliography	485
Index	487

INTRODUCTION

1

Overview	2
An Example	3
The Value of S+SEQTRIAL	5
S+SEQTRIAL Features	8
A Complete Software Environment	8
Stopping Rule Computation	8
Design Evaluation	9
Monitoring Clinical Trials	9
Analyzing and Interpreting Your Results	10
Using This Manual	11
Intended Audience	11
Organization	12
Typographic Conventions	12
Technical Overview	13
Product Website	14
Background Reading	14
Technical Support	15

OVERVIEW

Welcome to the *S+SEQTRIAL 2 User's Guide*.

S+SEQTRIAL is an S-PLUS software library for designing, monitoring, and analyzing clinical trials using *group sequential* methods. In a classical *fixed sample* design, the sample size is set in advance of collecting any data. The main design focus is choosing the sample size that allows the clinical trial to discriminate between the null and alternative hypotheses, thereby answering the scientific questions of interest.

A disadvantage of all fixed sample designs is that you always use the same number of subjects regardless of whether the true treatment effect is very beneficial, marginal, or actually harmful relative to the placebo. To address this problem, it is increasingly common to introduce interim analyses in order to ensure patient safety and efficient use of resources.

In a sequential design, data are monitored throughout collection, and a decision to stop a trial can be made before all of the data are accrued. In classical sequential studies, a test would be conducted after collecting every data point. The term *group sequential* refers to sequential studies in which the data are analyzed periodically, after a block of data is accrued. Group sequential designs are especially important for the design of Phase II and Phase III clinical trials, where ethical considerations such as patient safety and rapid approval of effective treatments are paramount. Indeed, the FDA now recommends group sequential studies in many cases.

The basic aspects of fixed sample design—specification of size, power, and sample size—are all present in group sequential design. The difference is that with group sequential tests, sample size is no longer a single fixed number. Instead, the design focus for group sequential tests is selecting a *stopping rule* defining the outcomes that would lead to early termination of the study for an appropriate schedule for interim analyses. In this way, the average number of subjects exposed to inferior treatments can be decreased, and the ethical and efficiency considerations of clinical testing are better addressed.

An Example

Let's look at an example. This manual teaches you how to use S+SEQTRIAL to design, analyze, and interpret a clinical trial like this one.

In a Phase III clinical trial to confirm the benefit of a new drug for the treatment of acute myelogenous leukemia, patients from the Memorial Sloan Kettering Cancer Center were randomly assigned with equal probability to receive either the new treatment (idarubicin) or the standard treatment (daunorubicin). The primary study objective was to demonstrate a difference in the rate of complete remission between the new and standard treatment.

A group sequential design was used for this trial with two interim analyses: one analysis after accruing 45 patients in each treatment arm, and a second analysis after accruing 65 patients. The maximal sample size for the trial was 90 patients in each treatment arm. The left panel of Figure 1.1 plots the stopping rules for this group sequential design. The design stopped at either of the two interim analyses if the new drug showed superiority or inferiority relative to the existing treatment. Otherwise, it concluded at the final analysis with a decision for superiority of the new treatment, inferiority of the new treatment, or an inability to declare that either treatment is better than the other (which might have been interpretable as approximate equivalence between the two treatments, depending on the minimal difference that was judged clinically important to detect).

For comparison, the right panel shows the fixed sample test with equivalent type I error and power as the group sequential test. The fixed sample test requires approximately 88 patients per arm, rather than the 90 patients per arm that would be accrued if the group sequential trial continued to the final analysis.

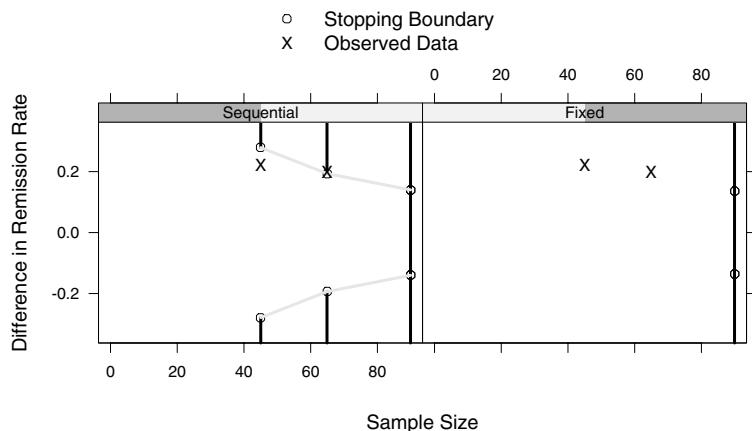


Figure 1.1: The stopping rules for a group sequential test to demonstrate a difference in the rate of complete remission between a new and standard treatment for acute myelogenous leukemia.

When the design phase was completed, the trial was begun. At the first analysis, 78% of the patients receiving idarubicin had complete remission as compared with 56% of the patients receiving daunorubicin. The difference in rates, 22%, was not statistically large enough to declare idarubicin better and the trial was continued. At the second analysis, patients receiving idarubicin still had a remission rate of 78% while those receiving daunorubicin had a rate of 58%. A difference in remission rates of 20% was statistically significant at the second analysis and the trial was stopped.

Figure 1.2 shows the average sample number (ASN) and power curves for both the group sequential test and the fixed sample test with equivalent size and power. The fixed sample test, which has a single analysis after accruing 90 patients in each treatment arm, would have taken considerably longer to complete.

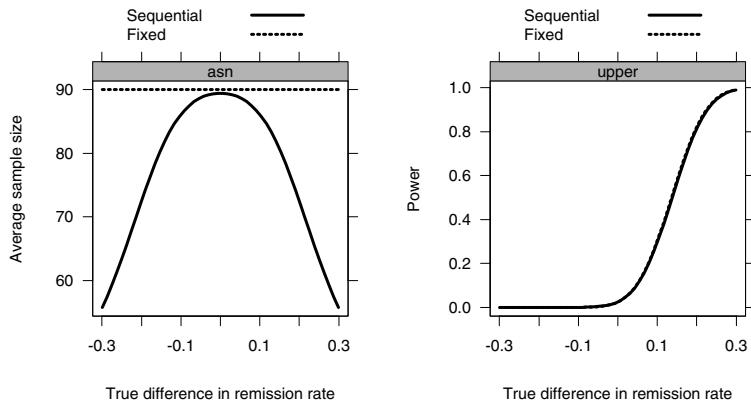


Figure 1.2: Left plot: the average sample number (ASN) for the group sequential design is substantially smaller than for the equivalent fixed sample test. Right plot: the power curves for the group sequential test and the fixed sample test are visually indistinguishable.

On average, group sequential designs require fewer subjects than equivalent fixed sample tests. For example, in the trial for treatment of myelogenous leukemia, the ASN for the group sequential test is potentially much lower than for the fixed sample test for (see Figure 1.2). This increase in efficiency comes essentially without cost: the maximum sample size for the group sequential test is the same as for the fixed sample test and the power curves are virtually identical.

The Value of S+SEQTRIAL

The difficulty introduced by interim analyses is that you need special methods and software. It is not appropriate to repeatedly apply a fixed sample test; doing so causes an elevation of the Type I statistical error. The sampling density for the test statistic is highly non-Gaussian due to the sequential nature of the test. (See Figure 1.3 for a typical density.) To adjust the stopping rules so that the test has the desired Type I statistical error, and to compute standard quantities such as power curves and confidence intervals, special software is needed to numerically integrate over such densities. S+SEQTRIAL performs these functions for you.

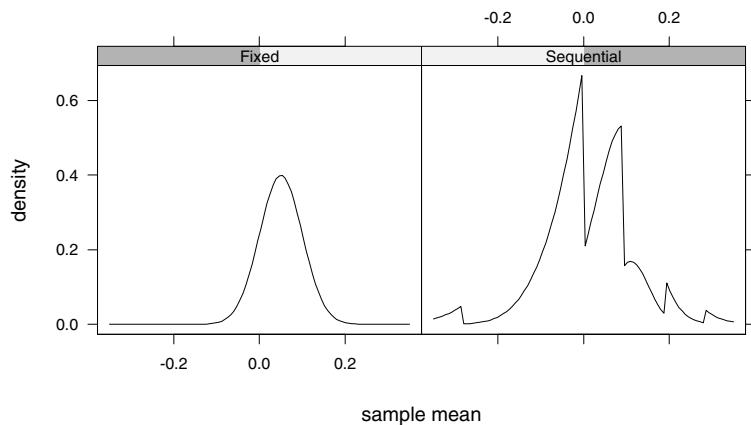


Figure 1.3: A typical sampling density for the test statistic for a fixed sample test (left) and group sequential test (right). The density for the group sequential test is highly non-Gaussian with discontinuities associated with the stopping boundaries.

Software is also needed for selecting and evaluating the most appropriate group sequential design. In comparison to fixed sample designs, group sequential tests offer much greater flexibility in the design of a clinical trial. The design parameters include not only power and sample size, but also:

- The number and timing of the analyses;
- The efficiency of the design;
- The criteria for early stopping (evidence against the null hypothesis, the alternative hypothesis, or both);
- The relative ease or conservatism with which a study is terminated at the earliest analysis versus later analyses.

S+SEQTRIAL helps you to select the best design.

Note that S+SEQTRIAL assumes that the treatment response is evaluated using a test statistic that is approximately normally distributed (for a fixed sample size), and that the increments of

information accrued between successive analyses can be reasonably regarded as independent. The vast majority of clinical trials are based on such statistical tests.

S+SEQTRIAL FEATURES

S+SEQTRIAL addresses all facets of the conduct of clinical trials: from design to monitoring to analysis. Here are some of the main features of this product.

A Complete Software Environment

S+SEQTRIAL offers a complete computing environment for applying group sequential methods, including:

- A fully object-oriented language with specialized objects (such as design objects, boundary objects, and hypothesis objects) and methods (such as operating characteristics and power curve plots);
- Full integration into the S-PLUS language for customized analyses, allowing you to extend S+SEQTRIAL as your applications demand;
- An intuitive graphical user interface oriented towards both the clinical trialist and the statistician (Windows version only);
- Many low-level routines for specialized analyses (for example, densities and quantiles);
- An open software design with well-defined building blocks;
- Easy comparative plots of boundaries, power curves, average sample number (ASN) curves, and stopping probabilities;
- User-selected scales for boundaries: sample mean, z-statistic, fixed sample p -value, partial sum, error spending, Bayesian posterior mean, and conditional and predictive probabilities;
- Publication quality graphics based on the powerful Trellis Graphics system (Cleveland, 1993; Becker & Cleveland, 1996).

Stopping Rule Computation

S+SEQTRIAL offers a variety of techniques for computing stopping rules, including:

- The unified family of group sequential designs described by Kittelson & Emerson (1999), which includes all common group sequential designs: Pocock (1977), O'Brien & Fleming

(1979), Whitehead triangular and double triangular (Whitehead & Stratton, 1983), Wang & Tsiatis (1987), Emerson & Fleming (1989), and Pampallona & Tsiatis (1994);

- A new generalized family of designs. S+SEQTRIAL includes a unified parameterization for designs, which facilitates design selection, and includes designs based on stochastic curtailment, conditional power and predictive approaches;
- Applications including normal, Binomial, Poisson, and proportional hazards survival probability models;
- Designs appropriate for one-arm, two-arm, and regression probability models;
- One-sided, two-sided, and equivalence hypothesis tests, as well as new hybrid tests;
- Specification of the error spending functions of Lan & DeMets (1989) and Pampallona, Tsiatis, & Kim (1993);
- Arbitrary boundaries allowed on different scales: sample mean, z-statistic, fixed sample p -value, partial sum, error spending, Bayesian posterior mean, and conditional and predictive probabilities;
- Exact boundaries computed using numerical integration.

Design Evaluation

S+SEQTRIAL includes a variety of techniques for evaluating designs, including:

- Power curves;
- Maximal sample size calculations;
- Sample size distributions: ASN curves and quantile curves;
- Stopping probabilities;
- Conditional power;
- Statistical inference at the boundaries;
- Bayesian properties (normal prior).

Monitoring Clinical Trials

S+SEQTRIAL offers a variety of techniques for monitoring trials, including:

- The error spending approach of Lan & DeMets (1989) and Pampallona, Tsitais, & Kim (1995);
- Constrained boundaries within the unified group sequential design family of Kittelson & Emerson (1999);
- Stochastic curtailment.

Analyzing and Interpreting Your Results

Finally, S+SEQTRIAL includes a variety of techniques for analyzing and interpreting your results, including:

- Inference based on analysis time ordering (Tsiatis, Rosner & Mehta, 1984) and sample mean ordering (Emerson & Fleming, 1990);
- Exact p -values;
- Exact confidence intervals;
- Point estimates adjusted for stopping rules: bias adjusted mean (Whitehead, 1986), median unbiased estimates, UMVUE;
- Bayesian posterior inferences (normal prior).

USING THIS MANUAL

This manual describes how to use the S+SEQTRIAL module, and includes detailed descriptions of the principal S+SEQTRIAL functions.

On the Windows platform, most S+SEQTRIAL functions can be run through dialogs available in the graphical user interface (GUI).

Functions can also be entered through the **Commands** window—the traditional method of accessing the power of S-PLUS. Under UNIX, this is the only way to use S+SEQTRIAL. Regardless of the platform, some advanced features are only available from the command line.

All discussions and examples in this manual are based on GUI input, but command line equivalents are given wherever possible, like this:

From the command line, type:

```
> seqDesign(power=0.8)
```

The command line equivalents are stored in help files, to make it easier for you to copy and paste them into S-PLUS. For example:

Commands for each chapter are stored in a help file:

```
> help(chapter3.seqtrial)
```

See Appendices A and B for a complete command line reference.

Intended Audience

Like the S+SEQTRIAL module, this book is intended for statisticians, clinical researchers, and other analysts involved in the design, monitoring and analysis of group sequential clinical trials. We assume a working knowledge of S-PLUS, such as can be obtained from reading your *S-PLUS User's Guide*.

We also assume that you have a basic knowledge of statistics and, in particular, are familiar with group sequential statistics. This book is not meant to be a text book in group sequential statistics; we refer you to the excellent books by Whitehead (1997) and Jennison & Turnbull (1999) listed in the Bibliography for recommended reading in this area.

Organization

The main body of this book is divided into twelve chapters which take you step-by-step through the S+SEQTRIAL module.

- Chapter 1 (this chapter) introduces you to S+SEQTRIAL, lists its features, and tells you how to use this manual and contact technical support;
- Chapter 2 shows you the basics of using S+SEQTRIAL, such as how to start and quit the program, how to launch the dialog system, and the typical workflow you follow;
- Chapter 3 contains a tutorial illustrating the use of S+SEQTRIAL;
- Chapter 4 covers the design process common to fixed sample and group sequential trials;
- Chapter 5 covers the design process specific to group sequential trials;
- Chapter 6 has instructions on evaluating and comparing group sequential designs;
- Chapter 7 discusses how to monitor a group sequential trial;
- Chapter 8 discusses issues in making inferences from a group sequential study, and reporting your results;
- Chapters 9-12 contain four detailed case studies. Working through these case studies is an excellent way to learn how to use S+SEQTRIAL.

This book also includes two appendices: appendix A provides a list of the S+SEQTRIAL functions organized into categories; appendix B contains individual help files for S+SEQTRIAL functions.

Typographic Conventions

This book uses the following typographic conventions:

- The *italic font* is used for emphasis, and also for user-supplied variables within UNIX, DOS, and S-PLUS commands.
- The **bold font** is used for UNIX and DOS commands and filenames, as well as for chapter and section headings. For example,

setenv S_PRINT_ORIENTATION portrait

SET SHOME=C:\S-PLUS

In this font, both “ and ” represent the double-quote key on your keyboard ("").

- The typewriter font is used for S-PLUS functions and examples of S-PLUS sessions. For example,

```
> seqDesign(power=0.8)
```

Displayed S-PLUS commands are shown with the S-PLUS prompt >. Commands that require more than one line of input are displayed with the S-PLUS continuation prompt +.

NOTE: Points of interest are shown like this.

Command line equivalents to dialog input are shown like this.

Online Version

This *S+SEQTRIAL User’s Manual* is also available online. It can be viewed using Adobe Acrobat Reader, which is included with S-PLUS. Select **Help ▶ Online Manuals ▶ SeqTrial User’s Manual**.

The online version is identical in content to the printed one but with some particular advantages. First, you can cut-and-paste example S-PLUS code directly into the **Commands** window and run these examples without having to type them. Be careful not to cut-and-paste the “>” prompt character, and notice that distinct colors differentiate between command language input and output.

Second, the online text can be searched for any character string. If you wish information on a certain function, for example, you can easily browse through all occurrences of it in the guide.

Also, contents and index entries in the online version are hot-links; click them to go to the appropriate page.

Technical Overview

An advanced supplement to this manual is also available online, and can be viewed using Adobe Acrobat Reader. The *S+SEQTRIAL Technical Overview* reviews the formal foundations of group sequential statistics, and establishes the standard notation used in this manual. Select **Help ▶ Online Manuals ▶ SeqTrial Technical Overview**.

**Product
Website**

For the latest news, product updates, a list of known problems, and other information on S+SEQTRIAL, visit the product website at

<http://www.insightful.com/products>

**Background
Reading**

For users familiar with S-PLUS, this manual contains all the information most users need to begin making productive use of S+SEQTRIAL. Users who are *not* familiar with S-PLUS, should read their *S-PLUS User's Manual*, which provides complete procedures for basic S-PLUS operations, including graphics manipulation, customization, and data input and output.

Other useful information can be found in the *S-PLUS Guide to Statistics*. This manual describes how to analyze data using a variety of statistical and mathematical techniques, including classical statistical inference, time series analysis, linear regression, ANOVA models, generalized linear and generalized additive models, loess models, nonlinear regression, and regression and classification trees.

For references in the field of group sequential statistics, see the bibliography at the end of this manual. The excellent books by Whitehead (1997) and Jennison & Turnbull (1999) may have valuable insight.

TECHNICAL SUPPORT

If you purchased S+SEQTRIAL in the last 60 days, or if you purchased a maintenance contract for S+SEQTRIAL, and you have any problems installing or using the product, you can contact S+SEQTRIAL technical support in any of the following ways:

North, Central, and South America

Contact Technical Support at Insightful Corporation:

Telephone: 206.283.8802 or 1.800.569.0123, ext. 235,
Monday-Friday, 6:00 a.m. PST (9:00 a.m. EST) to 5:00 p.m.
PST (8:00 p.m. EST)

Fax: 206.283.8691

Email: support@insightful.com

All Other Locations

Contact Insightful Corporation, European Headquarters at:

Christoph Merian-Ring 11, 4153 Reinach, Switzerland

Telephone: +41 61 717 9340

Fax: +41 61 717 9341

Email: support.ch@insightful.com

GETTING STARTED

2

Overview	18
Using S+SEQTRIAL	19
Starting and Quitting S+SEQTRIAL	19
Organizing Your Working Data	20
Getting Help	20
S+SEQTRIAL Graphical User Interface	22
The SeqTrial Menu Hierarchy	22
The S+SEQTRIAL Dialog	22
Typical Workflow	27

OVERVIEW

This chapter describes the following tasks particular to running the S+SEQTRIAL module:

- How to start and quit S+SEQTRIAL, including setting up your environment so that S+SEQTRIAL is started whenever you start S-PLUS;
- How to organize your working data in S+SEQTRIAL;
- How to get help;
- How to use the S+SEQTRIAL graphical user interface (GUI);
- The typical workflow to follow when using S+SEQTRIAL.

Some of the procedures in this chapter vary depending on whether you run S+SEQTRIAL under Windows.

USING S+SEQTRIAL

If you have used S-PLUS before, this chapter is a review of S+SEQTRIAL. If you have not used S-PLUS before, see the recommended reading list on page 14 to find out how to learn more about S-PLUS before proceeding.

Starting and Quitting S+SEQTRIAL

To start S+SEQTRIAL, first start S-PLUS:

- Under UNIX, use the command `Splus6` from your shell prompt.
- Under Windows, select **Start ▶ Programs ▶ S-PLUS 6**.

See your *S-PLUS User's Guide* for more detailed instructions on starting S-PLUS.

To add the S+SEQTRIAL menu hierarchy, dialogs, and functions to your S-PLUS session, choose **File ▶ Load Module**, then select `seqtrial` from the **Module** list box.

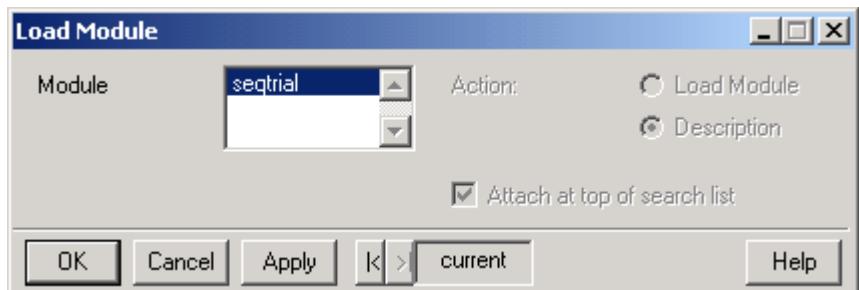


Figure 2.1: Sample **Load Module** dialog.

Click **OK**.

From the command line, add S+SEQTRIAL to your S-PLUS session by typing
`> module(seqtrial)`

If you plan to use S+SEQTRIAL extensively, you may want to customize your S-PLUS startup routine to automatically attach the S+SEQTRIAL module. You can do this by adding the line

`module(seqtrial)` to your `.First` function. If you do not already have a `.First` function, you can create one from the command line by typing:

```
> .First <- function() { module(seqtrial) }
```

From the command line, you can remove the S+SEQTRIAL module from your S-PLUS session by typing

```
> module(seqtrial, unload=T)
```

You can also create a `.S.init` file and add `module(seqtrial)` to it. Each time you start an S-PLUS session, the S+SEQTRIAL module is automatically loaded.

Organizing Your Working Data

To help you keep track of the data that you analyze with S+SEQTRIAL, you can create separate directories for individual projects. Each project level directory should have an S-PLUS `.Data` subdirectory.

The best way to organize multiple projects is by creating separate *project folders* and *chapters*. A project folder is used for storing the data and documents you create and modify during a session, and chapters are used for holding database objects. You can tell S-PLUS to prompt you for a given project before each upcoming session, and you can use chapters to attach databases for a session. Both projects folders and chapters automatically create `.Data` folders for you to hold working data. See Chapter 9, Working With Objects and Databases, in the *S-PLUS User's Guide* for more information about using project folders and chapters.

Getting Help

Context-sensitive help for the S+SEQTRIAL dialogs can be accessed by clicking the **Help** buttons in the various dialogs, by clicking the context-sensitive **Help** button on the toolbars, or by pressing the F1 key while S-PLUS is active.

S+SEQTRIAL also provides help files for virtually all S+SEQTRIAL functions. (Some functions intended for internal use have no help files.) Function help can be accessed by choosing **Help ▶ Available Help ▶ seqtrial**.

From the command line, you can obtain help on function someFunction by typing

> `help(someFunction)`

or

> `?someFunction`

The contents of the S+SEQTRIAL function help files are shown in Appendix B.

S+SEQTRIAL GRAPHICAL USER INTERFACE

This section describes the S+SEQTRIAL menu hierarchy and dialog system (Windows platform only).

The SeqTrial Menu Hierarchy

When you add the S+SEQTRIAL module to your S-PLUS session (see page 19), the multilevel **SeqTrial** menu is automatically added to your main S-PLUS menu bar. The **SeqTrial** menu hierarchy allows you to specify the basic clinical trial structure (the number of comparison groups) and a probability model, as shown in Figure 2.2.

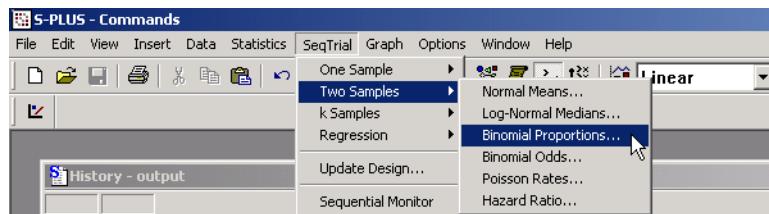


Figure 2.2: *SeqTrial* menu hierarchy

The contents of the **SeqTrial** menu hierarchy are described fully in Chapter 4.

The S+SEQTRIAL Dialog

The result of choosing any **SeqTrial** menu option is to launch the S+SEQTRIAL dialog. The fields in this dialog, and the meaning of the values you supply, vary somewhat depending on the model you specify in the menu hierarchy. For example, if you choose **SeqTrial** ► **Two Samples** ► **Binomial Proportions**, the S+SEQTRIAL dialog looks like that shown in Figure 2.3.

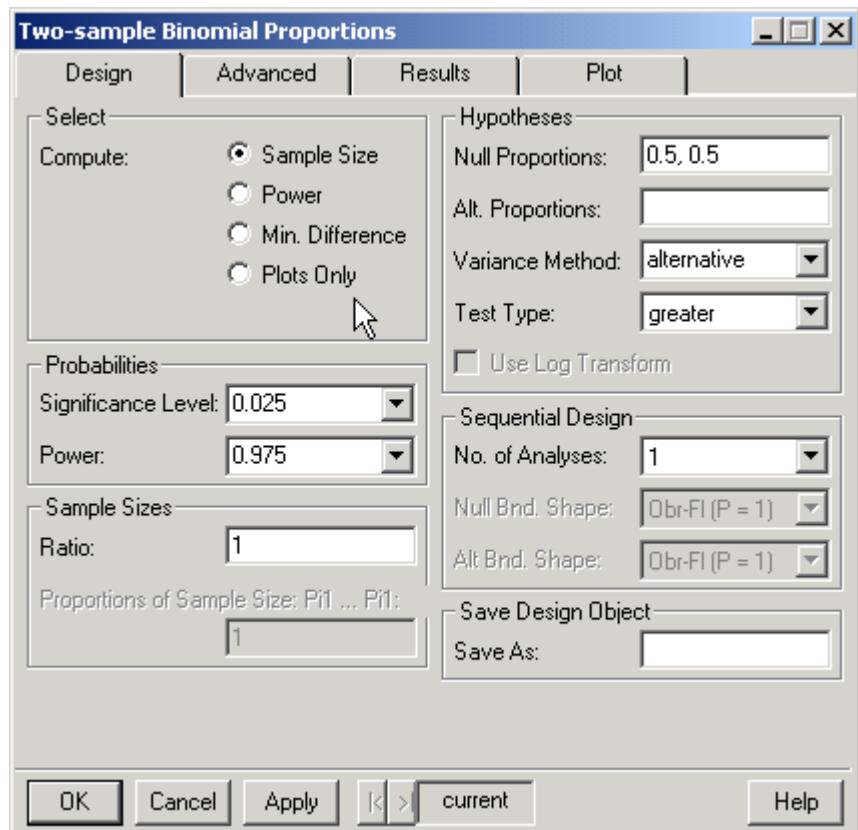


Figure 2.3: The default S+SEQTRIAL dialog for a two sample test of Binomial proportions.

Tabbed Pages

The S+SEQTRIAL dialog contains four tabbed pages.

- The **Design** tab contains design parameters common to both fixed sample and group sequential designs, as well as parameters for quickly specifying basic group sequential designs.
- The **Advanced** tab contains design parameters for advanced group sequential designs.
- The **Results** tab contains options for producing summary, power, and average sample size tables to evaluate your designs.

- The **Plot** tab contains options for producing plots to evaluate your designs.

To see the options on a different page of the dialog, click the page name, or press CTRL+TAB to move from page to page. When you choose OK or Apply (or press ENTER), any changes made on any of the tabbed pages are applied to the current design object.

The fields on these pages are fully described in Chapters 4-6.

Modeless Operation

The S+SEQTRIAL dialog, like all S-PLUS dialogs, is *modeless*—it can be moved around on the screen and remains open until you choose to close it. This means you can make changes in the dialog and see the effect without closing it. This is useful when you are experimenting with design changes to a trial and want to see the effect of each change.

The **OK**, **Cancel**, and **Apply** Buttons

When you are finished setting options in the S+SEQTRIAL dialog, you can choose the **OK**, **Cancel**, or **Apply** button.

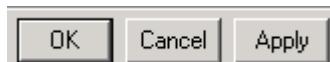


Figure 2.4: The **OK**, **Cancel**, and **Apply** dialog buttons.

Press the **OK** button or press ENTER to close the S+SEQTRIAL dialog and carry out the action.

Press the **Cancel** button or press ESC to close the S+SEQTRIAL dialog and discard any of the changes you have made in the dialog.

Sometimes changes cannot be canceled (for example, when changes have already been made with **Apply**).

The **Apply** button acts much like an **OK** button except it does not close the S+SEQTRIAL dialog. You can specify changes in the dialog box and then choose the **Apply** button or press CTRL+ENTER to see your changes, keeping the dialog open so that you can make more changes without having to re-select the dialog. If no changes have been made to the dialog since it was last opened or “applied,” the **Apply** button is greyed out.

Note: Choosing **OK** closes the dialog and executes the command specified in the S+SEQTRIAL dialog. If you do not wish the command to execute after the dialog closes, perhaps because you have already clicked on **Apply**, choose **Cancel** instead of **OK**.

The Dialog Rollback Buttons

The **Dialog Rollback** buttons let you restore the S+SEQTRIAL dialog to a prior state.



Figure 2.5: *The Dialog Rollback buttons.*

You can scroll back through each of the prior states until you find the set of values you want. Then you can modify any of these values and choose **Apply** or **OK** to accept the current state of the dialog. One use of **Dialog Rollback** is to restore a design object to a previous state.

Typing and Editing

The following tasks can be performed in the S+SEQTRIAL dialog using the special keys listed below.

Table 2.1: *Shortcut keys when using the S+SEQTRIAL dialog.*

Action	Special Keys
Move to the next option in the dialog	Tab
Move to the previous option in the dialog	Shift+Tab
Move between tabbed pages	CTRL+Tab

Table 2.1: Shortcut keys when using the S+SEQTRIAL dialog. (Continued)

Move to a specific option and select it	ALT+underlined letter in the option name. Press again to move to additional options with the same underlined letter.
Display a drop-down list	ALT+DOWN direction key.
Select an item from a list	UP or DOWN direction keys to move, ALT+DOWN direction key to close the list.
Close a list without selecting any items	ALT+DOWN direction key.

The S+SEQTRIAL dialog pages contains many text edit boxes. Text boxes allow you to type in information such as a file name or a graph title.

To replace text in a dialog:

1. Select the existing text with the mouse, or press ALT+underlined letter in the option name.
2. Type the new text.

Any highlighted text is immediately overwritten when you begin typing the new text.

To edit text in a text box:

1. Position the insertion point in the text box. If text is highlighted, it is replaced when you begin typing.
2. Edit the text.

Some text boxes allow input of vectors. When entering a vector, the individual elements are separated by commas.

TYPICAL WORKFLOW

The basic process of choosing a trial design involves defining a candidate design, evaluating the design's operating characteristics, then modifying the design as necessary to achieve the desired results.

More specifically, these are the typical steps to follow when using S+SEQTRIAL:

1. Use the **SeqTrial** menu hierarchy to specify the basic clinical trial structure and a statistical model (Chapter 4). This launches the S+SEQTRIAL dialog.
2. Use the **Design** tab of the S+SEQTRIAL dialog to specify design parameters common to both fixed sample and group sequential designs, such as the significance level and the null and alternative hypotheses (Chapter 4).
3. If you're designing a group sequential design, use the **Design** tab to specify basic group sequential parameters like the number and spacing of interim analyses and/or use the **Advanced** tab to specify advanced parameters like constraints (Chapter 5).
4. Use the **Results** and **Plot** tabs of the S+SEQTRIAL dialog to evaluate your design to see if it is truly appropriate for the setting (Chapter 6). Modify the design as necessary to achieve the desired results.
5. When you are satisfied with your design, you can begin to collect data. In a group sequential design, you must monitor the trial to determine if early stopping is appropriate (Chapter 7).
6. Make inferences and report your results at the conclusion of the trial (Chapter 8).

Command line equivalents for each step are given in the associated chapter. The corresponding commands are contained in a help file, for easier copying and pasting; e.g. the commands for Chapter 3 are available by typing

```
> help(chapter3.seqtrial)
```


TUTORIAL

3

Overview	30
Launching the S+SEQTRIAL Dialog	31
Starting S+SEQTRIAL	31
Selecting the Model	32
Designing a Fixed Sample Test	33
Specifying the Design	33
Evaluating the Design	36
Designing a Group Sequential Test	40
Specifying the Design	40
Evaluating the Design	42
Adjusting the Boundary Relationships	47
Advanced Group Sequential Designs	53
Where to Go From Here	56

OVERVIEW

To introduce you to the S+SEQTRIAL concepts, let's work through a simple two-sample study. The primary outcome of this study is 28-day mortality, and the estimated mortality on the placebo arm is 60%. The alternative hypothesis is that the mortality on the treatment arm is 40%. In terms of our comparison of the treatment arms, the null hypothesis is that the difference in the mortality rates is 0. The alternative is that the difference in the mortality rates (treatment minus control) is -0.2. Randomization is done equally to both groups.

This tutorial introduces you to S+SEQTRIAL, an S-PLUS software library for designing, monitoring, and analyzing clinical trials using group sequential methods. This tutorial guides you through:

- Starting S+SEQTRIAL;
- Using the S+SEQTRIAL menu hierarchy to open a new design;
- Specifying and evaluating a fixed sample design;
- Specifying and evaluating a group sequential design using O'Brien-Fleming boundary relationships with five equally spaced analyses;
- Adjusting the boundary relationships.;
- Specifying advanced group sequential designs.

Command line equivalents are given for each step in this tutorial.

This tutorial assume a basic familiarity with group sequential design concepts and the S+SEQTRIAL product, such as can be obtained by reading the MathSoft Technical Report *Group Sequential Design with S+SEQTRIAL*. More information on any of the steps described in this tutorial is contained in the *S+SEQTRIAL User's Manual*. (This tutorial is also contained in that document.)

On the Windows platform, most S+SEQTRIAL functions can be run through a dialog available in the graphical user interface (GUI). Functions can also be entered through the **Commands** window. Under UNIX, this is the only way to use S+SEQTRIAL. This tutorial assumes GUI input, but command line equivalents are given for all steps, like this:

```
> seqDesign(power=0.8)
```

LAUNCHING THE S+SEQTRIAL DIALOG

The first step in using S+SEQTRIAL is using the **SeqTrial** menu to launch the S+SEQTRIAL dialog appropriate for your clinical trial. This requires specifying the basic clinical trial structure (the number of comparison groups) and the probability model.

Starting S+SEQTRIAL

Add the S+SEQTRIAL module to your S-PLUS session by choosing **File ▶ Load Module**, then selecting **seqtrial** from the **Module** list. Click **OK**.

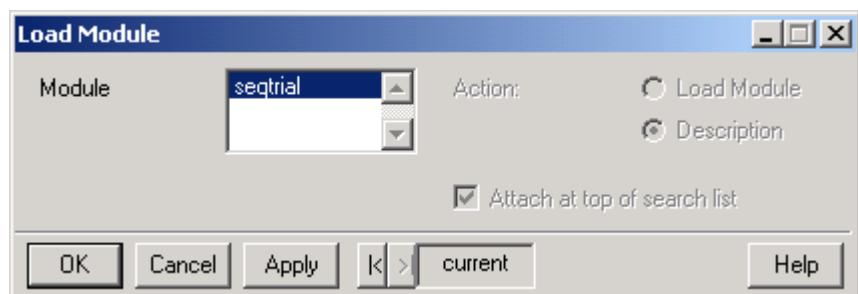


Figure 3.1: The **Load Module** dialog.

Chapter 2 contains more information on starting and quitting S+SEQTRIAL, including setting up your environment so that S+SEQTRIAL is started whenever you start S-PLUS.

From the command line, add the S+SEQTRIAL module to your S-PLUS session by typing

```
> module(seqtrial)
```

Note: All of the commands needed to run this tutorial are contained in a help file. Type:

```
> help(tutorial.seqtrial)
```

Selecting the Model

This is a placebo-controlled study, so there are two arms: the treatment arm and the control arm. The primary outcome is 28-day mortality. Because of this binary endpoint, and because we're modeling difference in proportions rather than odds, this study can be cast as a Binomial Proportions model.

Choose **SeqTrial ▶ Two Samples ▶ Binomial Proportions** to launch the appropriate S+SEQTRIAL dialog for this trial, as shown in Figure 3.2.

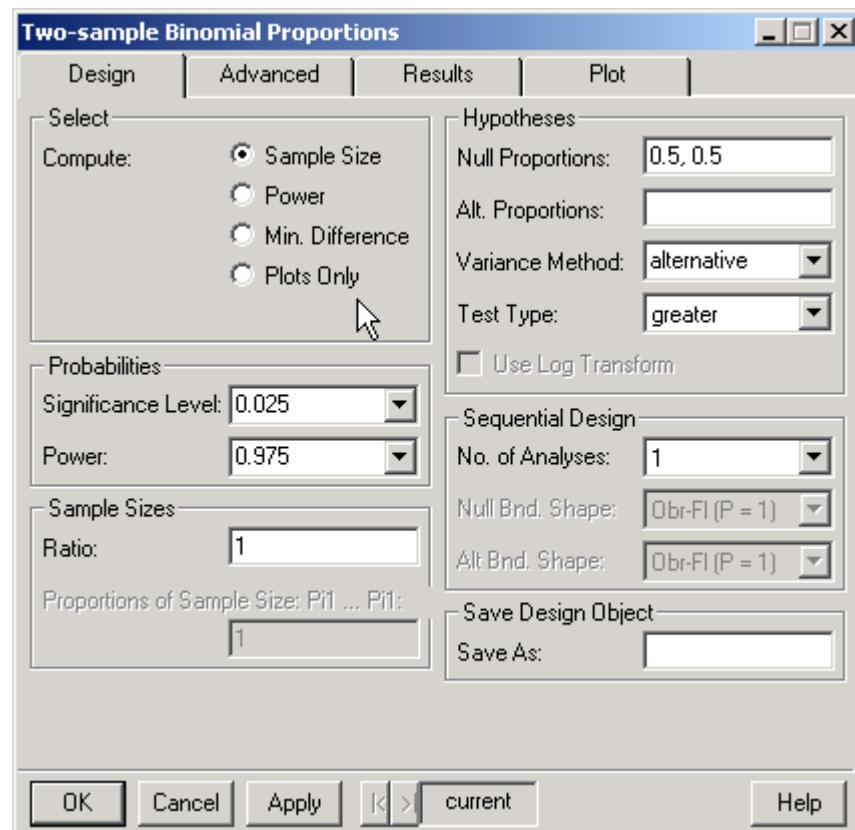


Figure 3.2: Using the S+SEQTRIAL menu to specify a two-sample test of binomial proportions.

The **SeqTrial** menu hierarchy and the probability models supported by S+SEQTRIAL are discussed in Chapter 2.

DESIGNING A FIXED SAMPLE TEST

First, let's design a classical fixed sample test. In a fixed sample design, the sample size is set in advance of collecting any data. Later we'll modify our design to have interim analyses.

Specifying the Design

The ultimate outcome of choosing any S+SEQTRIAL menu option is to launch the S+SEQTRIAL dialog. The **Design** tab of the S+SEQTRIAL dialog contains design parameters common to both fixed sample designs and group sequential designs. These parameters include the computational task, the sample size, the significance level, the power of a test, and the null and alternative hypotheses. We'll use the **Design** tab to design a fixed sample one-sided hypothesis test.

1. Indicate the computational task. There are three fundamental quantities which determine a design: the sample size, the power, and the minimum detectable difference between the means under the null and alternative hypotheses. Given any two of these quantities, S+SEQTRIAL can compute the third. Ensure that the **Sample Size** radio button (the default) is selected in the **Compute** group box.
2. Specify the **Probabilities**. By default the **Significance Level** is set to .025, in keeping with typical FDA recommendations for a one-sided test. The **Power** is set to .975, a choice that facilitates interpretation of negative studies.
3. For this study, randomization is done equally to the placebo and treatment groups, so ensure that the **Ratio** of the **Sample Sizes** between the two groups equals 1.0 (the default).
4. Specify the null and alternative hypotheses. In this study, the estimated mortality on the placebo arm is 60%, and the alternative hypothesis is that the mortality on the treatment arm is 40%. Enter **0.6,0.6** in the **Null Proportions** field, indicating the outcome on the treatment arm and the control arm under the null hypothesis. Enter **0.4,0.6** in the **Alt Proportions** field, indicating the outcome on the treatment arm and the control arm under the alternative hypothesis.

5. For this study, let's base our sample size computations on the variability of the test statistic under the alternative. Ensure that the **Variance Method** is set to `alternative` (the default).
6. Set the **Test Type** to `less`, indicating that you want a one-sided hypothesis test in which the difference in mortality rates (treatment group mortality minus comparison group mortality) is less under the alternative hypothesis (-0.2) than under the null hypothesis (0.0).
7. This is a fixed sample design, so ignore the **Sequential Design** group box. Later we'll modify our design to include interim analyses.
8. Save the design object under the name `tutorial.fix` using the **Save As** field.

The **Design** tab should now look like that shown in Figure 3.3.

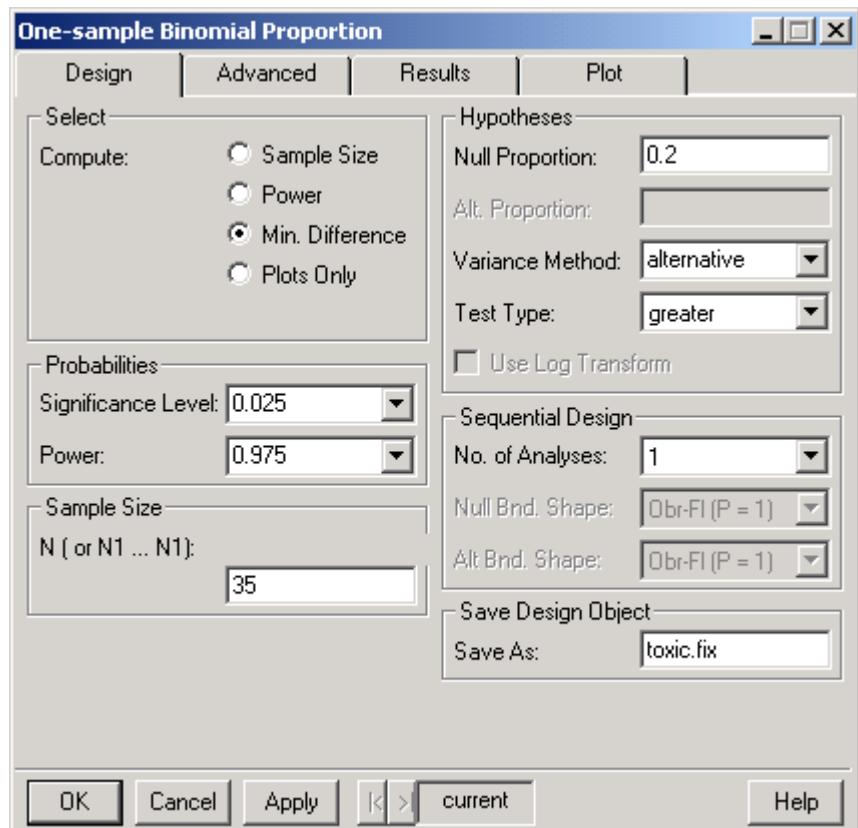


Figure 3.3: Sample S+SEQTRIAL dialog for a fixed sample design.

Specification of fixed sample designs is covered in Chapter 4.

From the command line, the same model can be selected and designed by typing:

```
> tutorial.fix <- seqDesign(prob.model="proportions",
+ arms=2, log.transform=F, size=.025, power=.975,
+ null.hypothesis= c(.6,.6),
+ alt.hypothesis=c(.4, .6), test.type="less",
+ variance="alternative")
```

Many of these arguments are the default values, so they could be omitted.

Evaluating the Design

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, log.transform =  
  F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,  
  0.6), variance = "alternative", test.type = "less", size  
  = 0.025, power = 0.975)  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
One-sided hypothesis test of a lesser alternative:  
  Null hypothesis : Theta >= 0      (size = 0.025)  
  Alternative hypothesis : Theta <= -0.2    (power = 0.975)  
(Fixed sample test)  
  
STOPPING BOUNDARIES: Sample Mean scale  
                  a      d  
Time 1 (N= 368.78) -0.1 -0.1
```

From the command line, you can display the same information about `tutorial.fix` using the `print` function:

```
> print(tutorial.fix)
```

The printed output may differ slightly between the versions created using menus and from the command line, in particular in the “Call” portion; one version may have arguments explicitly set to their default values, while the other omits them (implicitly setting them to the default values).

First, note that the `seqDesign` function is called by our fixed sample design. This is the same function that would be called in a group sequential design. The sole difference is that there is only one analysis specified in a fixed sample design, after all data have been accumulated. This single analysis is identified as `Time 1` in the printed summary information.

This design requires accrual of approximately 370 patients (185 per treatment arm). The boundary is displayed on the sample mean scale.

```
STOPPING BOUNDARIES: Sample Mean scale  
                  a      d  
Time 1 (N= 368.78) -0.1 -0.1
```

The critical values for this boundary indicate that if the estimated treatment effect (treatment group mortality minus comparison group mortality) is less than -0.1, then we will reject the null hypothesis of no treatment effect.

Let's also examine the decision boundary on the z-statistic scale.

1. Select the **Results** tab of the S+SEQTRIAL dialog.
2. Change the **Display Scale** from **Sample Mean** to **Z-Statistic**.

Click **Apply** to reprint the summary information.

Call:

```
seqDesign(prob.model = "proportions", arms = 2, log.transform =
F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,
0.6), variance = "alternative", test.type = "less", size
= 0.025, power = 0.975, display.scale = "Z")
```

PROBABILITY MODEL and HYPOTHESES:

```
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
    Null hypothesis : Theta >= 0      (size = 0.025)
    Alternative hypothesis : Theta <= -0.2   (power = 0.975)
(Fixed sample test)
```

```
STOPPING BOUNDARIES: Normalized Z-value scale
                  a      d
Time 1 (N= 368.78) -1.96 -1.96
```

From the command line, you can change the display scale using the update function

```
> update(tutorial.fix, display.scale="Z")
```

The boundary can also be displayed on another scale using the seqBoundary function

```
> seqBoundary(tutorial.fix, scale="Z")
```

or the changeSeqScale function

```
> changeSeqScale (tutorial.fix, outScale="Z")
```

The latter approach tends to be faster, because no new design is computed.

On this scale, the decision boundary corresponds to the familiar 1.96 critical value.

```
STOPPING BOUNDARIES: Normalized Z-value scale
                           a      d
Time 1 (N= 368.7818)    -1.96 -1.96
```

Reset the **Display Scale to Sample Mean**.

You can examine the power for a whole range of possible treatment effects by plotting the power curve.

1. Select the **Plot** tab of the S+SEQTRIAL dialog.
2. Select the **Power Curve** plot in the **Plots** groupbox.

Click **Apply** to generate the plot. Figure 3.4 shows the result.

From the command line, plot the power curve by typing:

```
> seqPlotPower(tutorial.fix)
```

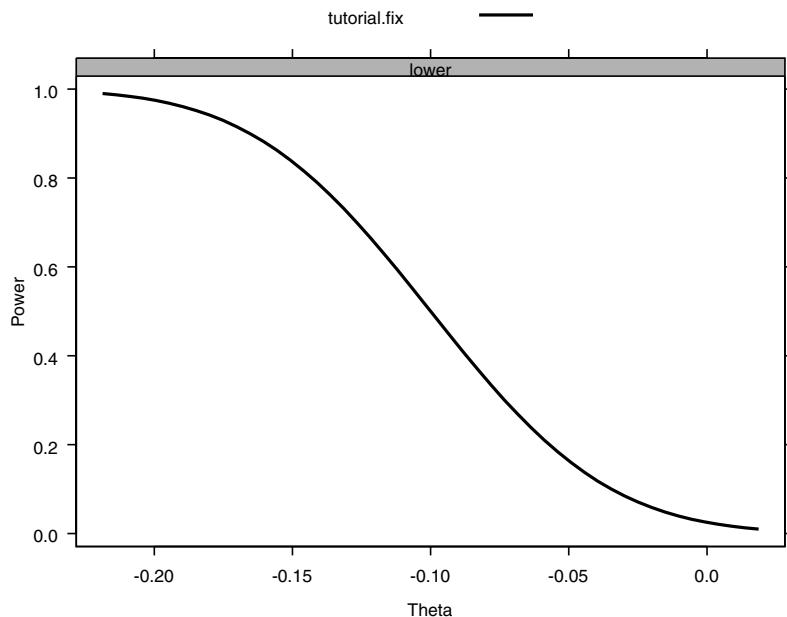


Figure 3.4: The power curves for a fixed sample one-sided test.

This plot displays the probability of exceeding the critical value, which is the power. From this plot, you can see that this sample size provides, say, 80% power to detect a treatment effect of a difference in mortality rates of about 0.14.

Design evaluation is described in more detail in Chapter 6.

DESIGNING A GROUP SEQUENTIAL TEST

A disadvantage of the design `tutorial.fix`, as in all fixed sample designs, is that it uses 370 subjects regardless of whether the true treatment effect is very beneficial, marginal, or actually harmful relative to the placebo. Let's modify our fixed sample design to be a group sequential design.

Specifying the Design

We'll define a one-sided symmetric group sequential design (Emerson & Fleming, 1989) using O'Brien-Fleming boundary relationships (O'Brien & Fleming, 1979) with five equally spaced analyses.

1. Click the **Design** tab. Basic group sequential designs can be specified from this page using the **Sequential Design** groupbox.
2. Click the **Interim Analyses** checkbox to specify a group sequential design.
3. Set the **Number of Analyses** to 5. This specifies the total number of analyses (interim plus final). The analyses are evenly spaced according to sample size.
4. Ensure that the **Boundary Shape** parameter P is set to **Obr-Fl (P=1)** (the default), which corresponds to O'Brien-Fleming boundary relationships. (The parameter P is discussed in more detail in Chapter 5 on page 118.) This implies that early stopping is possible under both hypotheses (to prevent early stopping under a hypothesis, set the corresponding boundary shape field to **No Early Stopping**).
5. Save the design object under the name **tutorial.obf** using the **Save As** field.

The **Design** tab should now look like that shown in Figure 3.5. Note that this design has the same size and power under the design alternative as the fixed sample design.

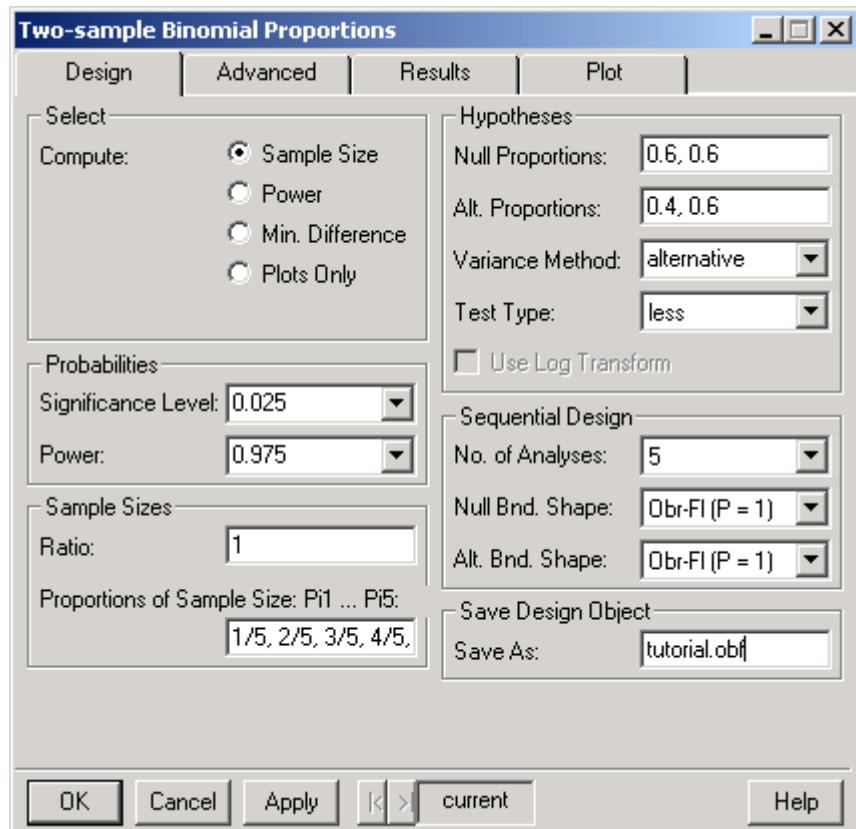


Figure 3.5: The S+SEQTRIAL dialog for a group sequential design with O'Brien-Fleming boundary relationships and five equally spaced analyses.

Group sequential design specification is discussed in Chapter 5.

From the command line, the same model can be selected and designed by typing:

```
> tutorial.obf <- update(tutorial.fix, nbr.analyses=5)
```

The O'Brien-Fleming boundary relationships ($P=1$) are used by default, as is early stopping under both alternatives.

Evaluating the Design Click **Apply** to create the design object and print out summary information in a report window:

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, log.transform =  
F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,  
0.6), variance = "alternative", nbr.analyses = 5,  
test.type = "less", size = 0.025, power = 0.975)  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
One-sided hypothesis test of a lesser alternative:  
Null hypothesis : Theta >= 0      (size = 0.025)  
Alternative hypothesis : Theta <= -0.2    (power = 0.975)  
(Emerson & Fleming (1989) symmetric test)  
  
STOPPING BOUNDARIES: Sample Mean scale  
a          d  
Time 1 (N= 77.83) -0.5000  0.3000  
Time 2 (N= 155.67) -0.2500  0.0500  
Time 3 (N= 233.50) -0.1667 -0.0333  
Time 4 (N= 311.34) -0.1250 -0.0750  
Time 5 (N= 389.17) -0.1000 -0.1000
```

From the command line, you can display the same information about `tutorial.obf` using the `print` function:

```
> print(tutorial.obf)
```

Note that a power curve plot is automatically generated when you click **Apply**. This is because the **Power Curve** plotting option is still selected on the **Plot** tab. This behavior of S+SEQTRIAL allows you to specify the evaluative plots you like, then to repeatedly regenerate them as you adjust your design specification. The same behavior applies to the different **Results** tables.

From the command line, you must explicitly call the `seqPlotPower` function to generate the new power curve.

Examining the printed summary information, we see that design tutorial.obf can stop early after accrual of $N_k = 78, 156, 234, 312$ subjects (39, 78, 117, or 156 per treatment arm). The maximal sample size, achieved when the trial continues to the final analysis, is 390, which is 5% bigger than the fixed sample design.

```
STOPPING BOUNDARIES: Sample Mean scale
      a      d
Time 1 (N= 77.83) -0.5000  0.3000
Time 2 (N= 155.67) -0.2500  0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000
```

Let's plot the average sample size for a range of possible treatment effects.

1. Select the **Plot** tab.
2. Deselect the **Power Curve** plot.
3. Select the **Average Sample Number** plot.

Click **Apply**.

From the command line, you can plot the average sample size curve using the seqPlotASN function:

```
> seqPlotASN(tutorial.obf, fixed=F)
```

The result is displayed in Figure 3.6.

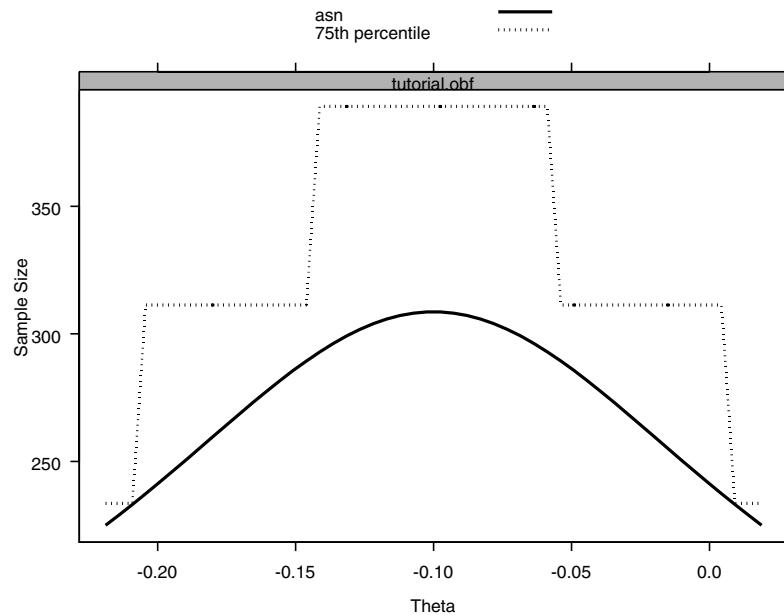


Figure 3.6: The average sample size (ASN) curve for the one-sided symmetric group sequential design using O'Brien–Fleming boundaries.

While the maximal sample size for the group sequential test is approximately 5% bigger than for the fixed test (390 vs. 370), the average sample size is considerably smaller, regardless of the true treatment effect. Note that there is an especially marked reduction in sample size in the cases where the true treatment effect is very beneficial or actually harmful relative to the placebo.

Now let's compare the power curves of the group sequential test to the fixed sample test:

1. Deselect the **Average Sample Number** plot.
2. Select the **Power Curve** plot.
3. In the **Designs to Plot** listbox, click the **Refresh** button, then select the fixed design object `tutorial.fix`.
4. Click on **Overlay Designs** in the **Options** groupbox.

The **Plot** tab should now look like that shown in Figure 3.7.

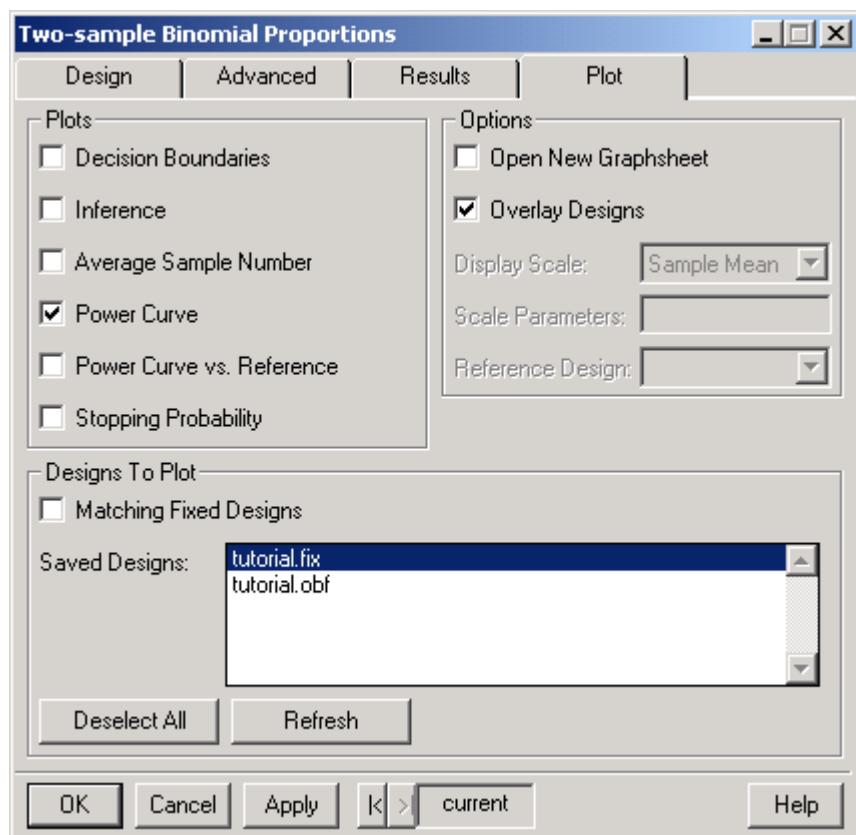


Figure 3.7: The **Plot** tab for comparing the power curves for group sequential and fixed sample designs.

Click **Apply**. Figure 3.8 shows the result. The power curves are visually indistinguishable.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(tutorial.ofb, tutorial.fix,
+               superpose.design=T)
```

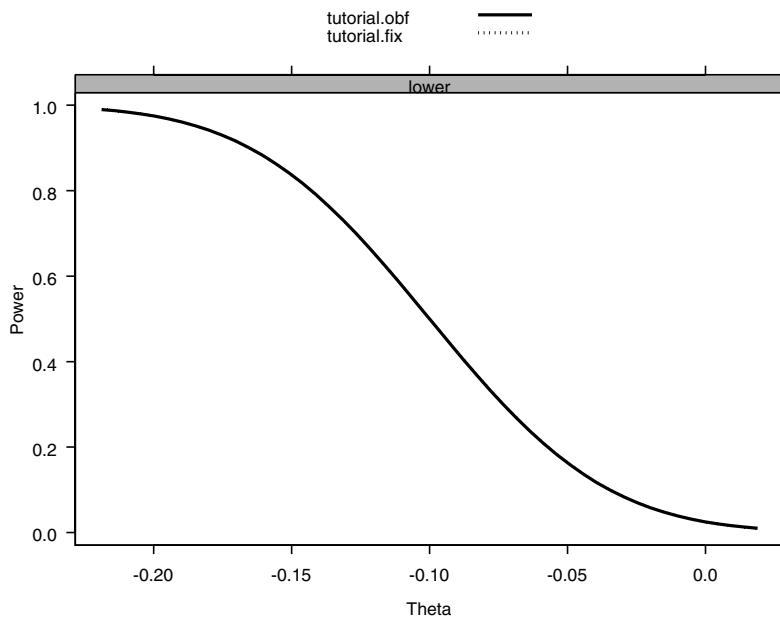


Figure 3.8: The power curves for the one-sided symmetric group sequential design using O'Brien-Fleming boundary relationships are visually indistinguishable from the power curves for the equivalent fixed sample test.

It is easier to compare the power curves by plotting the difference between them.

1. Deselect the **Overlay Designs** and **Power Curve** options.
2. Select the **Power Curve vs. Reference** plot type.
3. Set the **Reference Design** to **tutorial.fix**.

Click **Apply**. The result in this case is shown in Figure 3.9.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(tutorial.ofb, reference=tutorial.fix)
```

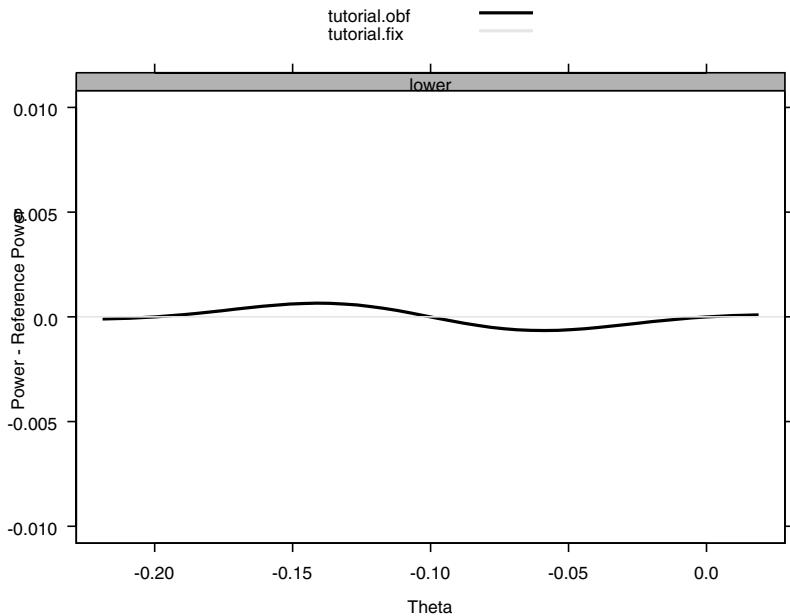


Figure 3.9: The difference curve created by subtracting the fixed design from the group sequential design.

You can see that the maximum difference in the two power curves is approximately .001.

Design evaluation is described in more detail in Chapter 6.

Adjusting the Boundary Relationships

Examining the printed summary information again, we see that the boundaries for design `tutorial.obf` are very conservative.

```
STOPPING BOUNDARIES: Sample Mean scale
      a      d
Time 1 (N= 77.83) -0.5000  0.3000
Time 2 (N= 155.67) -0.2500  0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000
```

The trial stops at the first analysis only if the estimated treatment effect (treatment group mortality rate minus comparison group mortality rate) is:

- less than -0.5, in which case the trial would cross the lower (or “a”) boundary and stop, and we would decide that the new treatment is beneficial;
- more than .3, in which case the trial would cross the upper (or “d”) boundary and stop, and we would decide that the new treatment is not sufficiently beneficial to warrant adoption.

One way to control the conservatism of a group sequential test is through the boundary shape parameter P . (The parameter P is discussed in more detail in Chapter 5 on page 118.) This parameter corresponds to the Wang–Tsiatis family of boundary relationships (Wang & Tsiatis, 1987). The design object `tutorial.obf` used a boundary relationship of $P=1$. Moving P towards zero creates a design that is less conservative at the earlier analyses. Let’s try this:

1. Deselect the **Power Curve vs. Reference** plotting option.
2. Select the **Design** tab.
3. Set both boundary shape parameters (**Null Bnd. Shape** and **Alt Bnd. Shape**) to **Pocock (P=.5)**, which corresponds to a one-sided symmetric design (Emerson & Fleming, 1989) with Pocock boundary relationships (Pocock, 1977).
4. Save the design object under the name **tutorial.poc**.

Click **Apply**.

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, log.transform =  
  F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,  
  0.6), variance = "alternative", nbr.analyses = 5,  
  test.type = "less", size = 0.025, power = 0.975, P = 0.5)
```

```
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
One-sided hypothesis test of a lesser alternative:  
  Null hypothesis : Theta >= 0      (size = 0.025)  
  Alternative hypothesis : Theta <= -0.2    (power = 0.975)  
(Emerson & Fleming (1989) symmetric test)
```

```
STOPPING BOUNDARIES: Sample Mean scale
      a      d
Time 1 (N= 108.15) -0.2236  0.0236
Time 2 (N= 216.31) -0.1581  -0.0419
Time 3 (N= 324.46) -0.1291  -0.0709
Time 4 (N= 432.61) -0.1118  -0.0882
Time 5 (N= 540.76) -0.1000  -0.1000
```

From the command line, the same model can be selected and designed by typing:

```
> tutorial.poc <- update(tutorial.obf, P=.5)
```

To print summary information, type

```
> print(tutorial.poc)
```

Again, the boundaries are displayed on the sample mean scale. This design stops at the first analysis with a decision that the treatment is beneficial if the estimated treatment effect (treatment group mortality rate minus comparison group mortality rate) is less than -.2236. It stops at the first analysis with a decision that the treatment is not sufficiently beneficial to warrant adoption if the estimated treatment effect is more than .0236 (that is, the mortality rate on the treatment arm is .0236 more than the mortality rate on the comparison arm).

Let's graphically compare the stopping rules for the two group sequential designs.

1. Select the **Plot** tab.
2. Select the **Decision Boundaries** plotting option.
3. Select the design object **tutorial.obf** from the **Designs to Plot** group box.
4. Deselect the **Overlay Designs** plotting option.

Click **Apply**. Figure 3.10 shows the result.

From the command line, you can compare the stopping rules for the two designs using the **seqPlotBoundary** function:

```
> seqPlotBoundary(tutorial.poc, tutorial.obf,
+     superpose.design=F)
```

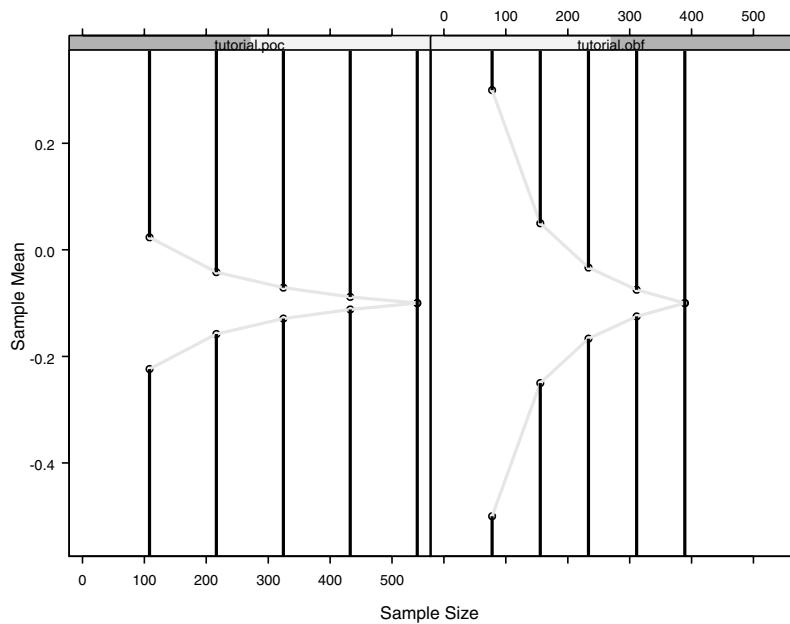
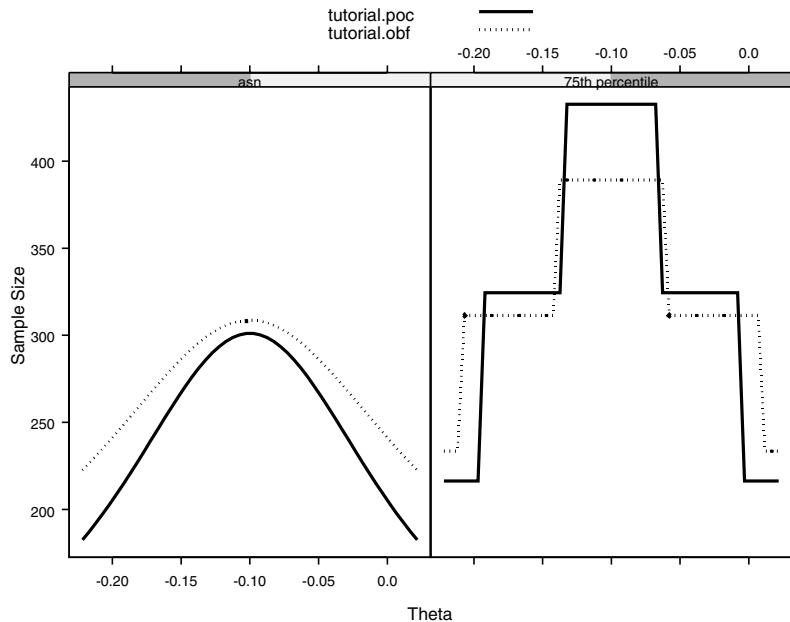


Figure 3.10: The left plot shows the Pocock decision boundaries for a one-sided group sequential design which are less conservative at the early analyses than the design with the O'Brien--Fleming boundary relationships (right).

The maximal sample size of `tutorial.poc` is 542, which is 39% larger than the maximal sample size of `tutorial.obf`. Let's graphically compare the ASN curves for the two group sequential designs:

1. Deselect the **Decision Boundaries** plot.
2. Select the **Average Sample Number** plot.
3. Ensure that the O'Brien-Fleming design object `tutorial.obf` is still selected in the **Designs to Plot** group box, then select the **Overlay Designs** plotting option.

Click **Apply**. Figure 3.11 shows the result.



From the command line, you can compare the ASN curves for the two designs using the `seqPlotASN` function:

```
> seqPlotASN(tutorial.poc, tutorial.obf, fixed=F)
```

Figure 3.11: Comparison of average sample size (ASN) curve for the designs corresponding to Pocock and O'Brien–Fleming boundary relationships.

You can see that while the maximal sample size for `tutorial.poc` is bigger than for `tutorial.obf`, the average sample size is uniformly smaller, at least over the range of alternatives displayed here. (For very extreme alternatives, a symmetric design with O'Brien-Fleming relationships would have a lower average sample size than a symmetric design with Pocock boundary relationships.)

In general, there is a tradeoff between efficiency and early conservatism of the test. When the alternative hypothesis is true, a design with Pocock (1977) boundary relationships tends to average a lower sample size than does a design that is more conservative at the earlier analyses. From the right panel of the Figure 3.11, however, you

can see that for many possible values of the true treatment effect, the larger maximal sample size of the design with Pocock boundary relationships results in a higher *75th* percentile for the sample size distribution than for the design with O'Brien-Fleming boundary relationships.

ADVANCED GROUP SEQUENTIAL DESIGNS

S+SEQTRIAL allows you to create most group sequential designs described in the statistical literature. For example, try creating a design using Whitehead's triangular test (Whitehead, 1992):

1. Select the **Design** tab.
2. Set the boundary shape parameters to **Triangular (P=1, A=1)**, which corresponds to Whitehead's triangular test.
3. Save the design object under the name **tutorial.tri**.
4. Click **OK**.

From the command line, the same model can be selected and designed by typing:

```
> tutorial.tri <- update(tutorial.obf, P=1, A=1)
```

You can create asymmetric tests as well. For example, let's define an asymmetric test which uses conservative rules for the upper boundary but more efficient rules for the lower boundary.

1. Choose **SeqTrial ▶ Update Design**.
2. Select **tutorial.obf** from the **Select Design** listbox, then click **OK**.
3. On the **Design** tab, set the **Alt Bnd. Shape** to **Pocock (P = .5)**
4. Save the design object under the name **tutorial.asymm**.
5. Click **OK**.

From the command line, the same model can be selected and designed by typing:

```
> tutorial.asymm <- update(tutorial.obf, P=c(.5,1))
```

An even larger family of advanced group sequential designs based on the unified family of group sequential designs (Kittelson and Emerson, 1999) can also be specified from the **Advanced** tab. For example, using the **Shape Parameters** groupbox on the **Advanced** tab you can specify the four boundary shape parameters (denoted P ,

R , A , and G ; see page 118) as vectors of length 4: one value for each of the four stopping boundaries a – d . For example, the previous design could be created using these steps:

1. Choose **SeqTrial ▶ Update Design**.
2. Select **tutorial.ofb** from the **Select Design** listbox, then click **OK**.
3. Save the new design under the name **tutorial.asymm**.
4. Select the **Advanced** tab and set the boundary shape parameter P to **.5,0,0,1**.
5. Click **OK**.

S+SEQTRIAL even lets you define hybrid tests that are in between one-sided and two-sided hypothesis tests. For instance, if you had an important secondary outcome, such as days in the intensive care unit (ICU), a treatment that was approximately equivalent with placebo with respect to 28-day mortality but superior with respect to days in the ICU might be of significant clinical interest. In such a case, the stopping boundary defined for the primary endpoint of 28-day mortality should consider that approximate equivalence of the new treatment and comparison groups might still be compatible with adoption of the new treatment.

Let's therefore design a study that has a lower boundary that stops early for clear superiority on the primary outcome, as measured by a decrease in mortality, and a upper boundary that stops early for lack of approximate equivalence between the treatment and comparison arms in the direction of the treatment arm being worse (having higher mortality).

1. Choose **SeqTrial ▶ Update Design**.
2. Select **tutorial.ofb** from the **Select Design** listbox, then click **OK**.
3. Save the new design under the name **tutorial.hybrid**.
4. Select the **Advanced** tab.
5. The epsilon shift parameter allows you to define hybrid tests between one-sided and two-sided tests. A vector of length two represents the upward shift of the lower hypothesis test and the downward shift of the upper hypothesis test. Set **Epsilon** to **1, .5**.

6. Click **OK**.

From the command line, the same model can be selected and designed by typing:

```
> tutorial.hybrid <- update(tutorial.obf,  
+     test.type="advanced", epsilon=c(1, .5))
```

This design is intermediate between the one-sided group sequential hypothesis tests examined above and a full two-sided hypothesis test, which would have an upper boundary designed to prove harm rather than lack of approximate equivalence.

WHERE TO GO FROM HERE

In this tutorial, you have covered the basics of using S+SEQTRIAL. The *S+SEQTRIAL User's Manual* following chapters systematically explores the same material in more depth. Of particular note are the four detailed case studies which illustrate the use of S+SEQTRIAL. Working through these analyses is the best way to learn how to use S+SEQTRIAL effectively.

SPECIFYING A FIXED SAMPLE DESIGN

4

Overview	58
Clinical Trial Design in S+SEQTRIAL	59
Probability Models	60
Statistical Hypothesis Tests	60
Determination of Sampling Scheme	63
Selection and Evaluation of Clinical Trial Designs	63
Types of Clinical Trials	64
One Sample	64
Two Samples	65
k Samples	66
Regression	66
Types of Probability Models	68
Normal Mean(s)	68
Log-Normal Median(s)	69
Binomial Proportion(s)	69
Binomial Odds	70
Poisson Rate(s)	70
Hazard Ratio	71
Creating a New Design	73
Examples	75
Updating an Existing Design	78
Design Parameters	80
Select	81
Probabilities	83
Sample Size(s)	85
Predictor Distribution (Regression Only)	90
Hypotheses	91
Sequential Design	97

OVERVIEW

This chapter describes the design features of S+SeqTrial for fixed sample tests. This chapter describes:

- The clinical trial structures available in S+SEQTRIAL;
- The types of probability models available in S+SEQTRIAL;
- How to use the **SeqTrial** menu hierarchy to create a new design;
- How to use the **SeqTrial** menu hierarchy to modify an existing design;
- How to specify fixed sample design parameters such as the computational task, the sample size, the significance level, the power of a test, and the null and alternative hypotheses.

Chapter 5 describes additional design parameters specific to group sequential designs.

CLINICAL TRIAL DESIGN IN S+SEQTRIAL

The process of designing a clinical trial typically involves refining vaguely stated scientific hypotheses into testable statistical hypotheses. This refinement of hypotheses includes

- Identifying groups to compare in order to detect the potential benefit of some treatment or prevention strategy.
- Specifying a probability model to describe the variability of treatment outcome and choosing some parameter to summarize the effect of treatment on the distribution of outcomes.
- Defining statistical hypotheses which the clinical trial will discriminate between.
- Defining the statistical criteria for evidence which will correspond to decisions for or against particular hypotheses.
- Selecting a sampling scheme which will allow sufficient precision in discriminating between the statistical hypotheses.

S+SEQTRIAL is designed to facilitate these stages of clinical trial design. You may choose from among clinical trials appropriate for single arm, two arm, and multiple arm studies, including the possibility of modeling continuous dose-response across multiple dose groups. The treatment outcome on each arm can be summarized by the mean, geometric mean, proportion or odds or response, or hazard rate. Available tests include the classical one-sided and two-sided hypothesis tests, as well as one-sided equivalence (non-inferiority) and two-sided equivalence tests. Both standard frequentist inference (based on estimates which minimize bias and mean squared error, confidence intervals, and p -values) and Bayesian inference (based on posterior probability distributions conditional on the observed data and a presumed prior distribution for the parameter measuring treatment effect) can be used to evaluate the clinical trial design. You may then use S+SEQTRIAL to compute the sample size requirements to provide the desired precision for inference when using either a fixed sample size or a stopping rule.

With the exception of the definition of the sampling scheme, the issues which need to be addressed when designing a clinical trial are the same whether the study is conducted under a fixed sample design

or with a group sequential stopping rule. In the remainder of this section, we first provide a brief overview of the issues involved in the selection of particular clinical trial design options and the general terminology and notation used by S+SEQTRIAL. In later sections of this chapter, we describe the mechanics of creating a clinical trial design in the context of fixed sample testing. The use of a stopping rule is addressed in Chapter 5.

Probability Models

A clinical trial is designed to investigate the effect that treatment has on some clinically important measure of disease. For instance, in the treatment of early cardiovascular disease, we might be interested in the effect of treatment on blood pressure, while when studying more advanced disease, we might be interested in the effect of treatment on length of survival.

We most often measure the effect of treatment according to some summary measure (e.g., the mean, median, proportion above some threshold, the hazard) of the distribution of clinical outcomes within a treated population. Ideally we compare the trend in outcomes between a population receiving an experimental treatment and a control population receiving some standard treatment.

Thus the often vaguely stated scientific hypotheses (e.g., the treatment will tend to reduce blood pressure) are restated as statistical hypotheses in terms of some parameter measuring treatment effect (e.g., the difference in mean blood pressure between treated and control populations). In S+SEQTRIAL, a number of different probability models can be specified to address particular measures of treatment effect. In each case, we use the notation θ to refer to the treatment effect parameter, which in S+SEQTRIAL output is written “Theta”. For some probability models you will specify the treatment effect parameter under each hypothesis by providing the hypothesized summary response for each group, rather than specifying the treatment effect parameter directly.

Statistical Hypothesis Tests

A clinical trial is designed to discriminate between two or more hypotheses, classically referred to as the “null” and “alternative” hypotheses. The null hypothesis $H_0: \theta = \theta_0$ typically represented the decision to be made in the absence of strong evidence to the contrary and generally corresponded to the status quo. Hence, when testing a new treatment the null hypothesis is most often chosen to correspond

to equality of response distribution between patients receiving the new treatment and patients receiving some comparison treatment (e.g., placebo or some standard therapy). For instance, in two arm studies, the typical choice is $\theta_0 = 0$, where θ measures a difference between summaries of response on individual treatment arms (e.g., θ is a difference in means or difference in proportions), and $\theta_0 = 1$ is the typical choice when θ measures a ratio between summaries of response on individual treatment arms (e.g., θ is an odds ratio or hazard ratio). In classical hypothesis testing, we usually control the type I error of rejecting the null hypothesis when it is in fact true. This “significance level” of the test is usually denoted by α . Thus, it can sometimes be regarded that the null hypothesis is that value of the treatment effect θ for which the probability of deciding for the alternative is α .

The alternative hypothesis is generally the situation which we are hoping to establish. In classical hypothesis testing, this is generally specified only by a direction. That is, when testing a null hypothesis that $H_0: \theta = \theta_0$, the alternative is classically stated merely as $H_1: \theta > \theta_0$ for a one-sided test of a greater hypothesis, as $H_1: \theta < \theta_0$ for a one-sided test of a lesser hypothesis, or as $H_1: \theta < \theta_0$ or $\theta > \theta_0$ for a two-sided test of a greater hypothesis. For the purposes of study design, however, a specific value θ_1 is chosen as the alternative for which the null hypothesis would be rejected with prespecified statistical power. In S+SEQTRIAL, we generally refer to such an alternative as θ_- for an alternative less than the null hypothesis and as θ_+ for an alternative greater than the null hypothesis.

Equivalence testing

The above description of statistical tests is based on the idea of establishing inequality of response distribution across groups. However, increasingly there are applications in which the purpose of a clinical trial is to ensure that the distribution of response for some new treatment is sufficiently close to that of some proven treatment that the two treatments could in some sense be regarded as equivalent. There are two common settings for such a clinical trial. In bioequivalence studies of a new formulation for a previously proven treatment, regulatory agencies such as the FDA may approve a new formulation based on clinical trials showing that bioavailability of the

active substance is within narrow limits of what is obtained with the previous formulation. Such “two-sided equivalence” trials correspond exactly to a two-sided hypothesis test, although the greatest emphasis would be on ensuring adequate power to reject the null hypothesis when the treatment difference is unacceptably high. In S+SEQTRIAL, the default value for power is equal to 1 minus the one-sided type I error, and thus by default a two-sided hypothesis test satisfies the power requirements necessary for a two-sided equivalence interpretation.

The second type of equivalence trial is the one-sided equivalence or noninferiority trial. Such a design is used when a new treatment is to be compared to an existing treatment that is known to be efficacious. When there exist secondary advantages to the new treatment, it may be acceptable to adopt the new treatment even when it shows some slight decrease in efficacy relative to the comparison treatment. This might be the case, for instance, when the comparison treatment has been previously found to confer a marked advantage over placebo. For example, if the comparison treatment was known to represent a two-fold increase in the median survival time over placebo, a new therapy that was 10% worse than the comparison treatment in that same population still confers significant survival advantage over placebo. In this setting, a one-sided equivalence trial might be used to establish that the new treatment was at least 90% as efficacious as the new treatment. We would design the clinical trial to ensure that we would have a sufficiently low probability of adopting the new treatment whenever the median survival time on the new treatment was truly less than 0.9 times that of a comparison group that received the existing active treatment. Hence, at the end of the trial we would be highly confident that the new treatment was noninferior (i.e., not unacceptably inferior).

S+SEQTRIAL refers to such noninferiority as “equivalence” trials. When specifying such a trial, the null hypothesis is specified according the null hypothesis of equality, and the alternative hypothesis is specified according to the maximal reduction in efficacy which would be regarded as “noninferior”. Clearly such trials can also be specified by using a one-sided hypothesis test and specifying the lower limit of acceptable efficacy as the null hypothesis.

Determination of Sampling Scheme

Different clinical trial settings often impose very different constraints on the design process. Classically, a major goal of statistical clinical trial design is to determine a sample size that provides the desired operating characteristics in terms of sufficiently low Type I error when the null hypothesis is true, as well as sufficiently high statistical power to detect some clinically important alternative. Ideally, the alternative hypothesis is chosen based on scientific and clinical issues, and the sample size is selected to meet those goals.

In practice, however, it is often the case that there are practical constraints on the sample size available. In such a situation, the process of clinical trial design is more one of either:

- Identifying the statistical power that the prescribed sample size provides to detect the clinically important alternative; or
- Identifying the alternative that can be detected with sufficiently high power using the prescribed sample size.

By default, when you specify an alternative hypothesis without explicitly requesting that the power be calculated, S+SEQTRIAL will compute the sample size which provides high power to detect the alternative. You may specify the level of significance and power at arbitrary levels, however S+SEQTRIAL by default will use level of significance and power settings that correspond to inference based on 95% confidence intervals.

Selection and Evaluation of Clinical Trial Designs

In order to address realistic situations, it is important that a software system for clinical trial design allow the user great flexibility in specifying the desired characteristics of the study. To that end, the S+SEQTRIAL dialog allows you to specify a design through a great many fields. Many of these arguments can be redundant, or even contradictory, if you are not careful. The aim of this chapter and the next is to guide you in the proper use of these fields.

The basic command line function for clinical trial design in S+SEQTRIAL is `seqDesign`. The `update` function can be used to modify an existing design object.

TYPES OF CLINICAL TRIALS

The first step in using S+SEQTRIAL to design a trial is to use the **SeqTrial** menu hierarchy to launch the appropriate S+SEQTRIAL dialog. This requires specifying:

- The basic clinical trial structure (the number of comparison groups);
- The probability model.

The types of clinical trial structures available in S+SEQTRIAL are discussed individually below. Probability models are described in the next section.

In conducting a clinical trial, you are performing an experiment to compare the response of subjects on a new experimental treatment or treatments to a comparison group of subjects receiving an existing or standard treatment. Evidence for a treatment effect is based on demonstrating that the distribution of outcomes on the treatment arms is different than that in the comparison group. There are several different ways that comparison groups can be structured in a clinical trial.

One Sample

In some trials, you may compare a single treatment group to historical controls. In the resulting **One Sample** (single arm) trial, the estimate of response is compared to some well-defined standard from the historical controls. While many Phase II cancer clinical trials are conducted using the idea of historical controls or some absolute standard for preliminary efficacy, problems with secular trends in patient populations or ancillary treatments mean that the patients entering the clinical trial may not be truly comparable to patients treated in the past.

In some clinical trials, each subject can serve as his or her own comparison while maintaining the rigor of an experimental design. For instance, in a crossover study of two treatments, each subject receives each treatment, with order of administration randomized to minimize confounding the measurement of treatment effect. Measures of treatment effect can then be made on each individual, then summarized across all subjects in the study.

Similarly, in a split plot design, treatments might be assigned to different areas of each subject's body. Many ophthalmological trials are conducted by assigning different treatments to each eye randomly, and then measuring the treatment effect in each individual. To the extent that carryover effects in the crossover study and other risk factors in a split plot design can be ignored, the analysis of such studies can proceed by collapsing the data to a single measurement on each individual and treating the trial as a **One Sample** trial.

In each of the above settings, the treatment effect θ is defined as some summary measure of response within a single sample, and estimates of treatment effect are compared to some absolute standard.

From the command line, specify this clinical trial structure using the `arms=1` argument to the `seqDesign` function.

Note: Advanced users can also use the **One Sample Normal Mean(s)** probability model to handle any test statistic that in large samples is normally distributed and has an independent increment structure. In this case, setting **Std. Deviation** to 1 on the dialog (or, equivalently, using the `variance=1` argument to the `seqDesign` function) will allow the sample size to represent the statistical information for the test statistic.

Two Samples

Most commonly, scientific standards dictate conducting a clinical trial with concurrent controls. When a new treatment at a single dose is compared to some standard therapy, this leads to a **Two Samples** (two arm) trial in which patients are randomized to receive one of two treatments. The treatment effect θ is defined as the comparison of the summary response between the two arms. The first arm is referred to as the *treatment* arm, and the second arm is referred to as the *comparison* arm. Differences in outcome between the arms are computed as the treatment summary response minus the comparison summary response. Ratios of response between the arms are computed as the treatment summary response divided by the comparison summary response.

From the command line, specify this clinical trial structure using the `arms=2` argument to the `seqDesign` function (the default).

k Samples

The two arm trial structure is easily generalized to a **k Samples** (multiple arm) trial in which multiple treatments are tested, often with one of the arms representing a control or standard therapy. The analysis of such a trial may be to compare each active treatment to the control treatment, or to perform all pair-wise comparisons.

S+SEQTRIAL allows fixed sample trials with multiple arms for the **Normal Means** and **Log-Normal Medians** probability models only (see page 68 for a description of these models):

- In a **Normal Means** model, the measure of treatment effect θ is the variance of mean response between the arms.
- In a **Log-Normal Medians** model, the measure of treatment effect θ is the variance of log median response between the arms. (The interpretation of the treatment effect as a median is based on the log-transformed response having a symmetric distribution. If this is not the case, the treatment effect parameter still can be interpreted as a comparison of geometric means or log geometric means.)

Note that S+SEQTRIAL does not handle sequential k arm trials.

From the command line, specify this clinical trial structure by setting the `arms` argument of the `seqDesign` function to k , where $k > 2$.

Regression

A **Regression** trial is a variant of a multiple arm trial in which different treatment groups receive varying doses of the same treatment. In this case, the group receiving a dose of 0 is the logical control group. The analysis of such a trial could be by analysis of variance, as described for a **k Samples** trial, but if you're primarily interested in detecting a dose-response relationship, you should consider a regression model for the dose.

S+SEQTRIAL allows **Regression** trials for the following probability models only (see page 68 for a description of these models):

- In a **Normal Means** model, a **Regression** trial corresponds to linear regression. The measure of the treatment effect θ is defined as the difference in mean response associated with a one unit difference in the predictor modeling the treatment level; in other words, θ is the regression slope for response against treatment level.
- In a **Log-Normal Medians** model, a **Regression** trial corresponds to linear regression of a log-transformed response. The measure of the treatment effect θ is defined as the ratio of median response (or median ratio) associated with a one unit difference in the predictor modeling the treatment level. (Optionally, you may specify the treatment effect as the log median ratio.) (The interpretation of the treatment effect as a median is based on the log-transformed response having a symmetric distribution. If this is not the case, the treatment effect parameter still can be interpreted as a comparison of geometric means or log geometric means.)
- In a **Binomial Odds** model, a **Regression** trial corresponds to logistic regression. The measure of the treatment effect θ is defined as the odds ratio (or, optionally, as the log odds ratio) associated with a one unit difference in the predictor modeling the treatment level.
- In a **Poisson Rates** model, a **Regression** model corresponds to Poisson regression. The measure of the treatment effect θ is defined as the rate ratio (or, optionally, as the log rate ratio) associated with a one unit difference in the predictor modeling treatment level.

From the command line, specify this clinical trial structure using the `arms=0` argument to the `seqDesign` function.

TYPES OF PROBABILITY MODELS

Probability models can be distinguished by several characteristics:

- The type of random variable. For example, the random variable might be continuous, discrete, or censored.
- The summary measure used to summarize the distribution of outcomes in each group. For example, the summary measure might be the mean, median (or geometric mean), proportion, odds, or hazard rate.
- The measure used for comparisons between groups. For example, groups might be compared by differences between summary measures or the ratio of summary measures.
- The method of quantifying statistical information.

The types of probability models available in S+SEQTRIAL are discussed individually below.

Normal Mean(s)

In many clinical trials, the response is some measurement that is assumed to be a continuously distributed random variable (for example, systolic blood pressure or tumor size). In S+SEQTRIAL such a model is called a **Normal Mean(s)** model, and this is the default. The outcomes on each treatment arm are summarized by the mean response. Comparisons among the treatment arms are reported as differences in means. Statistical information is quantified by the within-group variance.

A common assumption when using such endpoints is that sample sizes are sufficiently large that, by the central limit theorem, the sample mean within each treatment group can be regarded as having an approximately Gaussian distribution. For multiple arm trials, or cases in which regression is used, the assumption of equal variances among treatment groups must be satisfied. The distribution of the response under the null hypothesis is specified by providing an average response and the standard deviation within each group.

From the command line, specify this model using the `prob.model="normal"` argument to the `seqDesign` function.

Log-Normal Median(s)

The **Log-Normal Median(s)** model is the same as the **Normal Mean(s)** model, except that the data are logarithmically transformed before analysis. This model is typically used when the standard deviation of the response measurements is proportional to the mean.

The response is assumed to be a continuously distributed, skewed random variable (for example, serum cholesterol), which has a symmetric distribution after logarithmic transformation. The outcomes on each treatment arm are summarized by the median response (or, optionally, the log median response). Comparisons among the treatment arms are reported as ratios of medians (or, optionally, the log median ratio). Statistical information is quantified by the within-group variance of the log-transformed response.

Note: If the log transformed data can not be assumed to be symmetrically distributed, this probability model should be interpreted as comparisons of the geometric mean ratio (optionally the log geometric mean ratio), instead of the median.

From the command line, specify this model using the `prob.model="lognormal"` argument to the `seqDesign` function.

Binomial Proportion(s)

In many clinical trials, the response measures whether some event has occurred. Thus, the response is a binary (dichotomous) random variable having a value of 0 or 1. Examples of such endpoints might be death within two years or induction of a complete remission after treatment with a cancer chemotherapy. When sample sizes are sufficiently large, a normal approximation to the binomial distribution can be used to test hypotheses. Such models are called **Binomial Proportion(s)** models in S+SEQTRIAL.

The outcome son each treatment arm are summarized by the proportion of subjects having the response. Comparisons among the treatment arms are reported as differences in proportions. Statistical information is quantified by the mean variance relationship under the null or alternative hypothesis. This model is typically used only in one and two arm trials.

From the command line, specify this model using the `prob.model="proportions"` argument to the `seqDesign` function.

Binomial Odds

The **Binomial Odds** model is the same as the **Binomial Proportions** model, except that the outcomes on each treatment arm are summarized by the odds of the response, and comparison among the treatment arms are reported as odds ratios (or, optionally, as the log odds ratio). This model is appropriate when the data are analyzed using logistic regression.

From the command line, specify this model using the `prob.model="odds"` argument to the `seqDesign` function.

Poisson Rate(s)

Sometimes a clinical trial is planned with a response that counts the number of events occurring in some specified time period. Examples of such endpoints might be the number of asthma attacks suffered by patients during a six month period, or the number of hypoglycemic episodes experienced by patients with diabetes during a month. Typically, an underlying Poisson distribution is assumed for such measurements, and the results of the trial are then measured according to the event rate. When sample sizes are sufficiently large, a normal approximation to the Poisson distribution can be used to test hypotheses. Such models are called **Poisson Rate(s)** models in S+SEQTRIAL.

The outcomes on each treatment arm are summarized by the mean event rate. Comparisons among the treatment arms are reported as rate ratios (or, optionally, as the log rate ratio). Statistical information is quantified by the mean variance relationship under the null or alternative hypothesis. This model is appropriate when the data are analyzed using Poisson regression.

From the command line, specify this model using the `prob.model="poisson"` argument to the `seqDesign` function.

Hazard Ratio

Many clinical trials are conducted with the response reflecting the time to some event (for example, the time to transplant failure in patients receiving liver transplants). In such trials, it is usually the case that not all subjects will have had the event at the time of data analysis, so some observations will be *right censored*. A popular statistical model used to analyze such data is the proportional hazards model, which leads to the logrank test when comparing two samples. Such models are called **Hazard Ratio** models in S+SEQTRIAL.

Typically, there is no summarization of outcomes on individual treatment arms. Comparisons among the treatment arms are reported as hazard ratios. The null hypothesis generally corresponds to a hazard ratio of 1 between the groups. The approximate variance of the statistic is known from statistical theory without further specification from the user (it is proportional to the number of events). This model is appropriate when the data are analyzed using the logrank or Cox tests.

From the command line, specify this model using the `prob.model="hazard"` argument to the `seqDesign` function.

Note: The assumption that the time to event distribution would exhibit proportional hazards across treatment arms is a relatively strong one. Should this assumption be violated in the presence of a true treatment effect, the logrank test is still valid to test the null hypothesis of equality of time to event distributions, because under this null hypothesis the hazard functions are proportional (with a hazard ratio of 1). Thus the *p*-values reported for the logrank test are valid for testing the null hypothesis of equality of distributions. The interpretation of hazard ratio estimates are much more difficult in the setting of nonproportional hazards, however. Furthermore, the estimated standard errors for the hazard ratio estimate can be incorrect under nonproportional hazards alternatives, so in that setting the confidence intervals based on the proportional hazards assumption may not have the correct coverage probability.

Note that in **Hazard Ratio** models, statistical information about the treatment effect is roughly proportional to the number of events observed, rather than to the number of subjects. Therefore, S+SEQTRIAL computes the number of events needed to attain the desired power.

For trial planning purposes, however, it is necessary to estimate the number of subjects required to observe the computed number of events in a reasonable period of time. In order to calculate the number of subjects needed, some assumptions must be made about the underlying distribution of times to event and times to censoring. A common model is to assume that censoring occurs due to staggered study entry, and that subjects accrue to the study uniformly during the recruitment period. Under this model, subjects can be followed for events for some additional follow-up time following the accrual of the last subject to the study. For the purposes of estimating the number of subjects to accrue to the study, the distribution of times to event is assumed to be exponential, a reasonable approximation when the hazard rate is approximately constant over the study period.

S+SEQTRIAL provides a command line function `seqPHSubjects` to estimate the number of subjects to accrue using this model, and this function can also be accessed from the **Results** tab of the dialog for the proportional hazards model. For a specified number of events desired during a given accrual period and time of additional follow-up, you can estimate the number of subjects to accrue to the study by specifying the median time to event on the comparison arm, a hypothesized hazard ratio comparing the treatment arm to the comparison arm, and the ratio with which subjects will be randomized to the treatment versus comparison arms. Alternatively, the accrual rate can be specified along with either the accrual time or the follow-up time, and the unspecified time period will be computed.

CREATING A NEW DESIGN

Loading the S+SEQTRIAL module automatically adds the multilevel **SeqTrial** menu to your main S-PLUS menu bar. The first level of the **SeqTrial** menu hierarchy allows to you create a new design or to modify an existing design, as shown in Figure 4.1.

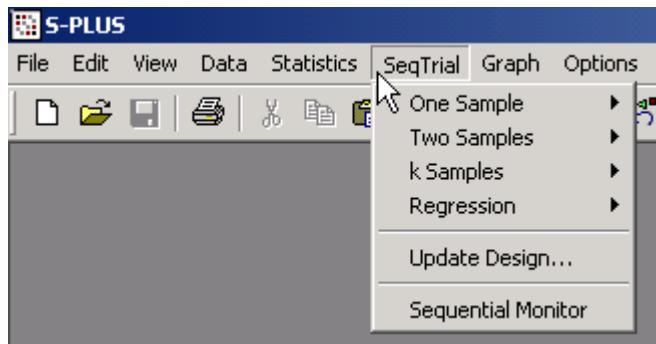


Figure 4.1: The first level of the **SeqTrial** menu hierarchy.

To create a new design, choose one of the trial structure options, as described on page 64. The second level of the S+SEQTRIAL menu hierarchy lists the available probability model types for your choice of trial structure. (See page 68.) For example, the result of choosing **SeqTrial ▶ Two Samples** is shown in Figure 4.2.

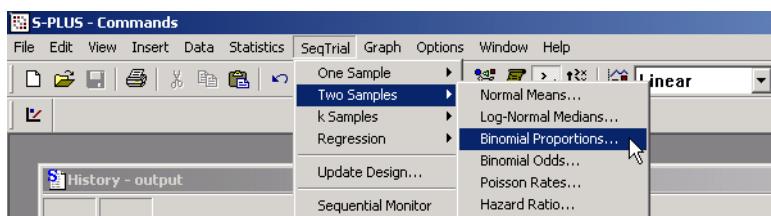


Figure 4.2: The result of choosing **SeqTrial ▶ Two Samples**, showing the available probability models for this trial structure.

Table 4.1 shows the probability models available for each trial structure.

Chapter 4 Specifying a Fixed Sample Design

Select the appropriate probability model to launch the S+SEQTRIAL dialog.

From the command line, create a new design object with the seqDesign function:

- Specify the number of comparison groups using the arms argument (arms=0 for regression).
- Specify the probability model using the prob.model argument.

For example, to specify a two-sample test of Binomial odds, type

```
> MyDesign <- seqDesign(arms=2, prob.model="odds")
```

Table 4.1: *The SeqTrial menu hierarchy.*

Trial Structure	Available Probability Models
SeqTrial ► One Sample ►	Normal Mean Log-Normal Median Binomial Proportion Binomial Odds Poisson Rate
SeqTrial ► Two Samples ►	Normal Means Log-Normal Medians Binomial Proportions Binomial Odds Poisson Rates Hazard Ratio
SeqTrial ► k Samples ►	Normal Means Log-Normal Means

Table 4.1: *The SeqTrial menu hierarchy. (Continued)*

Trial Structure	Available Probability Models
SeqTrial ► Regression ►	Normal Means Log-Normal Means Binomial Odds Poisson Rates

Examples

Let's look at some common clinical trial situations, and how the appropriate S+SEQTRIAL dialog would be launched using the **SeqTrial** menu hierarchy. For example:

- A Phase II clinical trial to investigate a new cancer chemotherapy with respect to its ability to shrink tumors (measured radiographically). Each subject has his or her tumor measured by CT scan at baseline, and then again after treatment. The primary endpoint is the difference in tumor size over the course of the study. The analysis is by a t-test using the difference in tumor size for each patient in the study. Choose **SeqTrial ► One Sample ► Normal Mean** to launch the appropriate S+SEQTRIAL dialog.
- A placebo controlled Phase III clinical trial to investigate a new treatment for high blood pressure. The primary endpoint measured on each subject is the subject's systolic blood pressure at the end of the study. The null hypothesis is that the difference in average responses between the two arms is 0. The analysis is by a t-test using the systolic blood pressure measurement at the end of treatment for each patient in the study. Choose **SeqTrial ► Two Samples ► Normal Means** to launch the appropriate S+SEQTRIAL dialog.
- A Phase II clinical trial to investigate the incidence of life-threatening leukopenia due to a new cancer chemotherapy. Each subject is categorized according to whether he or she experiences extremely low white blood cell counts during treatment with the chemotherapeutic agent. The analysis is by

a Z-test for a binomial proportion. Choose **SeqTrial ▶ One Sample ▶ Binomial Proportion** to launch the appropriate S+SEQTRIAL dialog.

- An alternative analysis of the above Phase II clinical trial to investigate the incidence of life-threatening leukopenia due to a new cancer chemotherapy might be based on a comparison of the odds of leukopenia. The analysis is based on a Z-test for the log odds ratio. Choose **SeqTrial ▶ One Sample ▶ Binomial Odds** to launch the appropriate S+SEQTRIAL dialog.
- A placebo controlled Phase III clinical trial to investigate a new treatment for sepsis. The primary endpoint for the study is the 28 day mortality rate, so each subject is categorized according to whether he or she is still alive 28 days after treatment. The null hypothesis is that the difference in 28 day mortality average responses between the two arms is 0. The analysis is by the Z-test for comparing binomial proportions. Choose **SeqTrial ▶ Two Samples ▶ Binomial Proportions** to launch the appropriate S+SEQTRIAL dialog.
- A Phase III clinical trial to investigate a new intensive treatment regimen for diabetes. To ensure that the rate of serious hypoglycemic episodes is not excessive, researchers count the number of such episodes for each subject over a three month period. The null hypothesis is that the difference in event rates between the two arms is 0. The analysis is by a log linear model comparing Poisson rates (Poisson regression with a binary predictor). Subjects are randomized to the new treatment or to the standard therapy group. Choose **SeqTrial ▶ Two Samples ▶ Poisson Rates** to launch the appropriate S+SEQTRIAL dialog.
- A placebo controlled Phase III clinical trial to investigate the ability of estrogen replacement therapy to prolong survival in post-menopausal women. For each woman, researchers measure the time between start of treatment and death from any cause. At the time of analysis there are many women still alive, so statistical methods for censored time to event data are used. Women are randomized to treatment and placebo groups, and hypotheses and boundaries are expressed in

terms of the hazard ratio. Choose **SeqTrial ► Two Samples ► Hazard Ratio** to launch the appropriate S+SEQTRIAL dialog.

UPDATING AN EXISTING DESIGN

Loading the S+SEQTRIAL module automatically adds the multilevel **SeqTrial** menu to your main S-PLUS menu bar. The first level of the **SeqTrial** menu hierarchy allows you to create a new design or to modify an existing design.

To modify an existing design, choose **SeqTrial ▶ Update Design**, as shown in Figure 4.3.

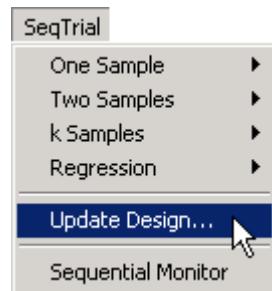


Figure 4.3: Using the *SeqTrial* menu hierarchy to update an existing design.

This launches the **Update Design** dialog shown in Figure 4.4

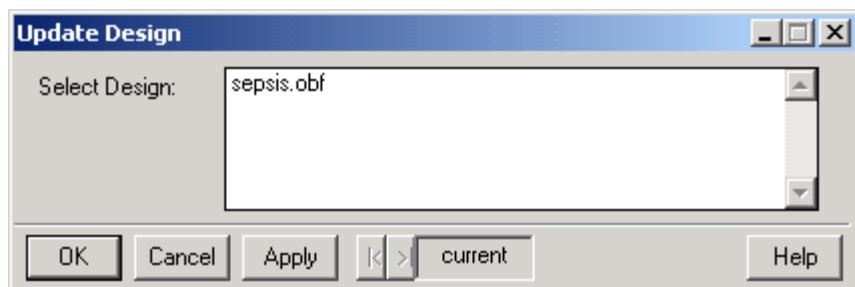


Figure 4.4: Sample *Update Design* dialog.

Select the design you wish to modify from the **Select Design** listbox and click **OK**. This restores the S+SEQTRIAL dialog for your previously defined design object.

From the command line, modify an existing design with the `update` function. This function takes as its first argument a design object, then updates the object according to subsequent arguments. When you want to update an existing design by choosing default values for arguments that had previously been specified, you can supply the `update` function with the named argument followed by an equal sign and no value.

For example, suppose you were conducting a one arm study in which the outcomes were summarized by the mean response. Further suppose you were interested in a one-sided level 0.10 test of the null hypothesis that the mean response is 0, and you want the test to have 90% power to detect the greater alternative of a mean response of 0.05. The following command line code would define such a design (note that we leave many arguments to take on their default values):

```
> MyDesign <- seqDesign(arms=1, alt.hypothesis= 0.05,  
+ size= 0.1, power= 0.9)
```

If you now wanted to update the design to take the default level of significance (which would be 0.025 for this one-sided test of a greater alternative) and to have 95% power, you could use the `update` function to “unspecify” the `size` argument and to supply the new value for the `power` argument as follows:

```
> MyNewDesign <- update(MyDesign, size=, power= 0.95)
```

DESIGN PARAMETERS

The ultimate outcome of choosing any **SeqTrial** menu option is to launch the S+SEQTRIAL dialog. The **Design** tab of the S+SEQTRIAL dialog contains design parameters common to both fixed sample designs and group sequential designs, as discussed below. These parameters include the computational task, the sample size, the significance level, the power of a test, and the null and alternative hypotheses.

The fields on the **Design** tab, and the meaning of the values you supply, varies somewhat depending on the probability model you specify. For example, the default **Design** tab for a two-arm test of Binomial proportions is shown in Figure 4.5.

The **Design** tab also contains parameters for quickly specifying basic group sequential designs; see Chapter 5 for more information on group sequential design.

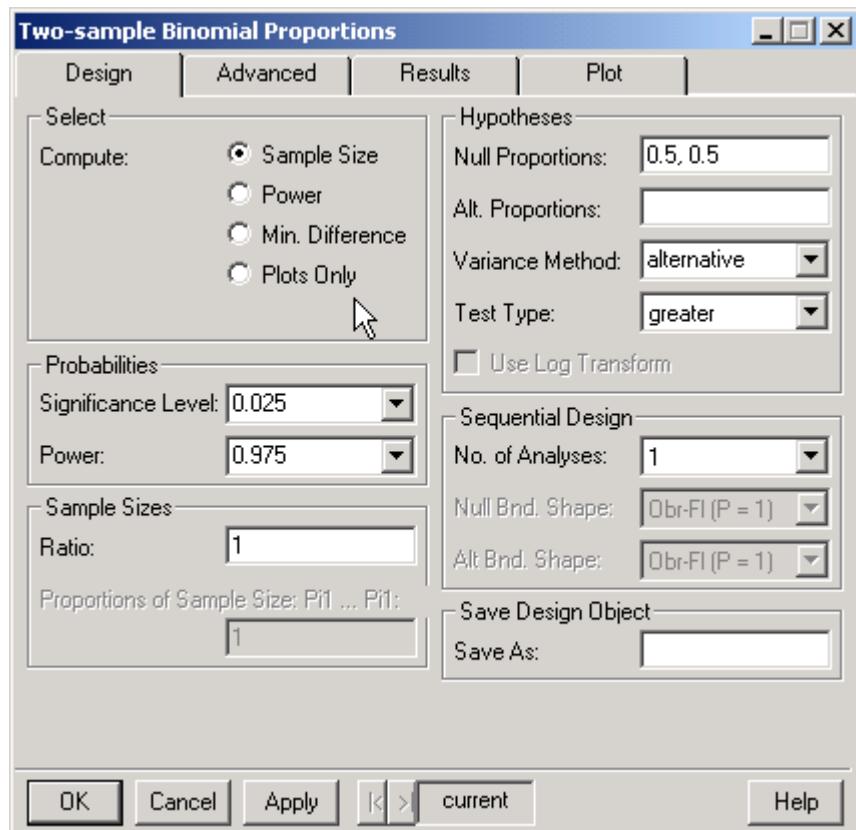


Figure 4.5: The **Design** tab for the **Two-sample Binomial Proportions** dialog.

The various fields of the **Design** tab are described below.

Select

The **Select** groupbox allows you to indicate the computational task. There are three fundamental quantities that determine a design: the sample size, the power, and the minimum detectable difference between the means under the alternative hypotheses. Given any two of these quantities, S+SEQTRIAL can compute the third.

To specify the desired computational task, click on the appropriate radio button in the **Select** groupbox:

- The **Sample Size** option computes the sample size to achieve a desired power for a specific alternative. You must specify the power and the alternative.
- The **Power** option computes the power that a given sample size provides to detect a specific alternative. You must specify the sample size and the alternative.
- The **Min. Difference** option computes the alternative for which a given sample size has the desired power. You must specify the sample size and the power.
- The **Plots Only** option makes no computation. This is useful for comparing previously defined designs using the **Plot** tab. No design object is created, and all fields on the **Design** tab are disabled.

From the command line, specify the computational task using the `sample.size`, `alt.hypothesis`, and `power` arguments to the `seqDesign` function. Two of these three arguments must have a numeric value, and the third will be computed by the function.

The `power` argument has a default numeric value based on the value of the `size` argument, which specifies the level of significance (Type I error) for the study. The `sample.size` argument has a default numeric value of 1, corresponding to a returned stopping rule on the standardized scale sometimes used in the statistical literature. The `alt.hypothesis` argument has no default numeric value. Thus, if you do not explicitly supply any of these arguments, `seqDesign` will compute the (standardized) alternative for which the study would have the default power with a maximal sample size of 1.

Continued on the next page...

If you do not specify a numeric value for `alt.hypothesis`, the numeric values of `sample.size` and `power` (either as specified by you or as taken from default values) are used to calculate the alternative hypothesis for which a clinical trial with the desired sample size will have the correct power. For one-sided hypothesis tests, the alternative will be computed according to whether the test is specified as being against a greater or less alternative hypothesis (see the documentation for argument `test.type`). For a two-sided hypothesis test, the alternative will be computed for the upper alternative (greater than the null hypothesis) unless you specify that `alt.hypothesis= "lower"`, in which case the alternative will be computed for the lower alternative.

If you do specify a numeric value for `alt.hypothesis`, and the value of `power` is numeric, any numeric value in the argument `sample.size` is used only to specify the relative spacing of interim analyses. The `seqDesign` function will compute the sample size which will provide the desired power to detect the specified alternative.

If you want to compute the power for which a study with a specific sample size can detect a specified alternative, you should specify `power= "calculate"` along with the desired value for `alt.hypothesis` and `sample.size` (if you do not want the maximal sample size to be 1).

Probabilities

The **Probabilities** groupbox allows you to specify the significance level and power of your design.

Significance Level

A fixed sample or group sequential design is usually constructed so that the resulting hypothesis test has an acceptable Type I error. The **Significance Level** field allows you to specify the level of Type I

error—that is, the probability of falsely rejecting the null hypothesis. The level of significance is either one-sided or two-sided according to the test type. The default value is 0.025.

From the command line, specify the significance level using the `size` argument to the `seqDesign` function.

When using the command line, the default value is 0.025 for one-sided tests (when the `test.type` argument to the `seqDesign` function is “less” or “greater”), and the default value is 0.05 for two-sided tests (when the `test.type` argument to the `seqDesign` function is “two.sided”).

Power

The **Power** field allows you to specify the desired power of the study to be used in the calculation of the sample size or the alternative hypothesis. The value for **Power** is taken to be the power of a lower hypothesis test if any of the following are specified in the **Hypotheses** groupbox (see page 91):

- **Test Type** indicates a one-sided hypothesis test of a `less` alternative;
- **Test Type** is set to `two.sided`, and the **Alternative** hypothesis is a numeric value less than the **Null** hypothesis;.
- **Test Type** is set to `two.sided` (`lower`), indicating a two-sided hypothesis test with a lower alternative used for power. (This option is only available when the computational task is set to **Min. Difference** in order to compute the alternative for which a given sample size has the desired power.)

Otherwise, the value for **Power** is taken to be the power of the upper hypothesis test. The default value is 0.975, which corresponds to a design in which the Type I and Type II statistical errors are equal. This ensures that a confidence interval computed at the end of the clinical trial using a 95% level of confidence will discriminate between the null and alternative hypotheses. This allows an interpretation of a negative study—a failure to reject the null hypothesis is tantamount to a rejection of the design alternative.

The **Power** field is disabled if you are computing the power (as specified in the **Select** groupbox).

From the command line, specify the power using the `power` argument to the `seqDesign` function. The value for `power` is taken to be the power of a lower hypothesis test if:

- `test.type= "less"`, indicating a one-sided hypothesis test of a lesser alternative;
- `test.type= "two.sided"`, indicating a two-sided hypothesis test, and `alt.hypothesis` is a numeric value less than the null hypothesis;
- `test.type= "two.sided"`, indicating a two-sided hypothesis test, and `alt.hypothesis= "lower"`.

Otherwise, the value for `power` is taken to be the power of the upper hypothesis test. The default value depends on the value of the `size` argument in such a way as to have one-sided Type I and Type II errors equal.

Sample Size(s)

In S+SEQTRIAL, the interpretation of the sample size depends on the probability model used of the clinical trial. In all probability models except the **Hazard Ratio** model, the sample size refers to the total number of subjects to be accrued to the study (across all arms). In the **Hazard Ratio** model, the sample size refers to the number of events observed. (See page 71 for more information.)

In order to accommodate unequal randomization to treatment arms, S+SEQTRIAL, allows the specification of a randomization ratio in **Two Samples** and **k Samples** designs, and it allows the specification of predictor distributions in **Regression** designs.

Ratio (Two Samples and k Samples designs only)

If you are *computing* the sample size, use the **Ratio** field to specify the ratio of sample sizes to be accrued to the trial arms in multi-sample models (**Two Samples** or **k Samples**). Enter a vector of length equal to the number of arms indicating the relative sample sizes per arm, or enter the ratio of the sample sizes on each arm relative to the comparison group. When entering more than one value, separate individual entries with commas. (If fewer values are entered than the

number of arms, the vector is expanded by appending 1's). The default ratio is 1, signifying equal sample sizes randomized to each arm.

N (or NI, ..., NI)

If you are computing the power or the alternative, you must specify a sample size. How you specify the sample size(s) depends on the basic clinical trial structure:

- In a **One Sample** design, enter a numeric value.
- In a **Two Samples** design, enter the total sample size in the **N** field, and use the **Ratio** field to specify the ratio of the treatment arm to the control arm. The default ratio is 1.
- In a **k Samples** design, enter the number of samples (k) in the **No. of Samples** field. Use the **N** field to enter sample sizes as a vector of length k (with individual values separated by commas), or enter the size of the combined sample size across all arms, then use the **Ratio** field to specify the ratios of the sample sizes on each arm. (The default ratio is 1, indicating equal sample sizes on each arm.)
- In a **Regression** design, enter a numeric value.

From the command line, use the `seqDesign` argument `sample.size` and `ratio` arguments to specify the sample size.

Results Tab: Accrual Design (Hazard Ratio models only)

(This section makes reference to some aspects of study design that are discussed in detail in later sections of this chapter. This topic is covered here due to its relationship to the interpretation of sample size requirements in time to event analyses.)

In the **Hazard Ratio** model, the sample size described above refers to the number of events observed. For trial planning purposes, however, it is necessary to estimate the number of subjects required to observe the computed number of events in a reasonable period of time. S+SEQTRIAL provides estimates of the number of subjects to accrue using a model in which subjects are entered onto the clinical trial uniformly over some accrual period, and then the analysis is

conducted after some follow-up time following accrual of the last subject. This model approximates the time to event distribution according to an exponential model.

Parameters for the accrual model are entered on the **Results Tab** of the **Hazard Ratio** model in the **Accrual Time** groupbox as displayed in Figure 4.6.

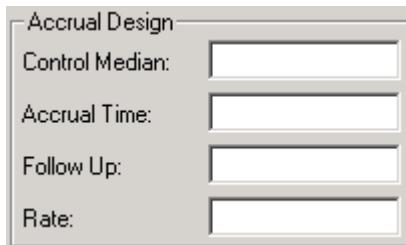


Figure 4.6: *Results tab for the Two-sample Hazard Ratio dialog showing the Accrual Design groupbox used to estimate the number of subjects to accrue to a time to event study in order to obtain the required number of events.*

The accrual model is specified by supplying the median time to event of the control group along with two of the three parameters specifying the accrual process. Each argument will accept a vector representing possible choices for the corresponding parameters. Any time unit may be used when entering numeric values in this groupbox, so long as it is consistent across all entries.

- In the **Control Median** text box enter a numeric vector indicating the median time to event for the comparison group.
- In the **Accrual Time** text box enter a numeric vector indicating candidate time(s) during which subjects will be accrued to the study. Entering this value presumes that you are not constrained by maximal accrual rates. (Only two of the three parameters of **Accrual Time**, **Follow-up**, and **Rate** should be entered.)
- In the **Follow-up** text box enter a numeric vector indicating the candidates for additional time(s) for which subjects will be followed for events following the end of the accrual period. (Only two of the three parameters of **Accrual Time**, **Follow-up**, and **Rate** should be entered.)

- In the **Rate** text box enter a numeric vector indicating the average number of subjects accrued during a single unit of time. Entering this value presumes that you are not constrained by maximal time available for accrual rates. (Only two of the three parameters of **Accrual Time**, **Follow-up**, and **Rate** should be entered.)

From the command line, use the seqPHSubjects or seqPHSampleSize functions to estimate the number of subjects to be accrued under various accrual models.

For example, suppose a two sample time to event study were designed as a one-sided level 0.025 test of the null hypothesis of a hazard ratio of 1.0 between treatment and comparison groups, and the sample size is to be chosen to provide 97.5% power to detect a true treatment effect corresponding to the greater hypothesis of a hazard ratio of 0.8, using a total of four analyses. Further suppose that it is planned that accrual should take place over a 4 year period, with an additional 2 years of follow-up following completion of accrual. We want to explore the way that accrual requirements will vary when the median time to event is 2 or 3 years on the comparison arm. Figure 4.6 shows the way the accrual model parameters would be entered into the S+SEQTRIAL dialog to obtain the following output in the **Report** window.

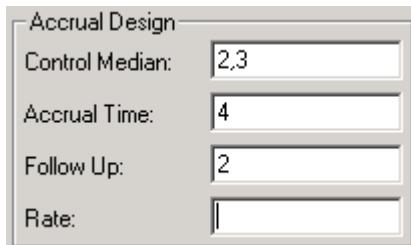


Figure 4.7: Results tab for the Two-sample Hazard Ratio dialog showing the Accrual Design groupbox, for a particular example.

PROBABILITY MODEL and HYPOTHESES:

Two arm study of censored time to event response variable

Theta is hazard ratio (Treatment : Comparison)

One-sided hypothesis test of a lesser alternative:

Null hypothesis : Theta >= 1 (size = 0.025)

Alternative hypothesis : Theta <= 0.8 (power = 0.975)

(Emerson & Fleming (1989) symmetric test)

STOPPING BOUNDARIES: Sample Mean scale

a	d
---	---

Time 1 (N= 322.37) 0.6400 1.2500

Time 2 (N= 644.74) 0.8000 1.0000

Time 3 (N= 967.11) 0.8618 0.9283

Time 4 (N= 1289.48) 0.8944 0.8944

Scalar input values:

accrualTime = 4

followupTime = 2

Calculated rate:

controlMedian=2	controlMedian=3
-----------------	-----------------

hazardRatio=1.0	441.9076	547.4327
-----------------	----------	----------

hazardRatio=0.8	466.3737	585.8569
-----------------	----------	----------

From this output, you can see that to provide 97.5% power to detect a hazard ratio of 0.8, you would need to observe 1290 events. Given an accrual time of 4 years and an additional follow-up period of 2 years, if the median time to event is 2 years on both arms, the accrual rate would need to average 441.9 persons each year (approximately 1767 subjects total). If the alternative hypothesis (a lower hazard rate on the treatment arm) is true the accrual rate would need to be higher in order to observed the necessary total number of events, with an average rate of 466.4 persons. Similar interpretations can be made for the case in which the median time to event for the comparison group were 3 years.

From the command line type:

```
> myDesign <- seqDesign(prob.model = "hazard",
+   alt.hypothesis = 0.8, test.type = "less",
+   nbr.analyses = 4)
> subjects <- seqPHSubjects(myDesign,
+   controlMedian = c(2,3),
+   accrualTime = 4, followupTime = 2)
> summary(subjects)
```

In addition, to obtain additional detail on the estimate times for the interim analyses do:

```
> print(subjects)
```

	accrualTime	followupTime	rate	hazardRatio	controlMedian
1	4	2	441.9076	1.0	2
2	4	2	466.3737	0.8	2
3	4	2	547.4327	1.0	3
4	4	2	585.8569	0.8	3

	nSubjects	analysisTimes.1	analysisTimes.2	analysisTimes.3
1	1767.630	2.326576	3.480820	4.513063
2	1865.495	2.366119	3.524533	4.559173
3	2189.731	2.472342	3.638693	4.675272
4	2343.428	2.501838	3.670157	4.706436

	analysisTimes.4
1	6.000000
2	6.000000
3	6.000000
4	6.000000

Predictor Distribution (Regression Only)

The **Predictor Distribution** groupbox, shown in Figure 4.8, only appears on the **Design** tab for **Regression** designs.

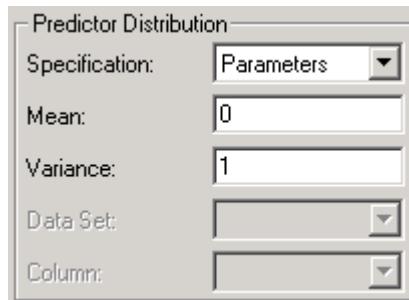


Figure 4.8: The **Predictor Distribution** groupbox, which only appears on the **Design** tab for **Regression** models.

You can specify the predictor distribution in two ways, as determined by the setting of the **Specification** field:

- If the **Specification** is set to **Parameters**, enter the **Mean** and **Variance** of a normal distribution from which the predictors are sampled.
- If the **Specification** is set to **Data Set**, enter the **Data Set** and **Column** that represent the sample of predictors that might be used in the clinical trial.

From the command line, the distribution of predictors for regression models is specified using the `ratio` argument.

Hypotheses

The **Hypotheses** groupbox allows you to specify such parameters as the null and alternative hypotheses, the variability, and the test type (one-sided or two-sided).

Null

The **Null** field allows you to specify the null hypothesis. The exact interpretation depends on the model type and whether a log transform is being performed.

How you specify the null hypothesis depends on the basic clinical trial structure:

- In a **One Sample** model, enter a scalar value summarizing the outcome for the single arm under the null hypothesis.
- In a **Two Samples** model, enter a vector of length two representing the summary measure of the outcome on the treatment arm and the control arm (in that order). If only a single value is entered, that value is presumed to be the summary measure of outcome on the treatment arm and it is assumed that the same value would be obtained on the comparison arm.
- In a **k Samples** model, the **Null** field is disabled, because the between group variance is always assumed to be 0.
- In a **Regression** model, you have two choices. You can enter a scalar value, in which case it is assumed to be the summary of outcome common to all values of the predictor under the null, or you can enter a vector of length two, in which case the first value is the summary of outcome for a population having the average predictor value, and the second value is the null value for a comparison between two populations differing in their predictor values by a single unit.

The default value depends upon the model type and whether a log transform is being performed: 0 for a **Normal Mean(s)** model, and 0.5 for a **Binomial Proportion(s)** model; the default is 1 for all other models, unless a log transform is specified, in which case the default is 0.

From the command line, use the `null.hypothesis` argument to the `seqDesign` function to specify the null hypothesis.

Alternative

If you are computing the sample size or the power, you must specify the nature of the hypothesis being tested. Ideally, the alternative hypothesis represents the minimum treatment difference that is clinically important to detect. The **Alt** field allows you to enter a numeric value specifying the alternative hypothesis. How you specify the alternative hypothesis varies depending on the basic clinical trial structure, in much the same way as the for the null hypothesis (see above):

- In a **One Sample** model, enter a scalar value summarizing the outcome for the single arm under the alternative hypothesis.
- In a **Two Samples** model, enter a vector of length two representing the summary measure of the outcome on the treatment arm and the comparison arm (in that order) under the alternative hypothesis. If only a single value is entered, that value is presumed to be the summary measure of outcome on the treatment arm under the alternative hypothesis, and it is assumed that the value for the comparison arm is the same as specified for the comparison arm under the null hypothesis.
- In a **k Samples** model, enter either a scalar value representing the between group variance, or a vector of group means or medians of length k .
- In a **Regression** model, enter a scalar value that is the alternative value for a comparison between two populations differing in their predictor values by a single unit.

From the command line, use the `alt.hypothesis` argument to the `seqDesign` function to specify the alternative hypothesis.

Standard Deviation/Variance Method

This field allows you to specify the variability associated with an observation. The name of this field and the interpretation of its argument depend on the type of model you are designing, as described below.

- **Normal Mean(s)** and **Log-Normal Median(s)** models

The variability field is labelled **Std. Deviation**, and refers to the standard deviation of a single observation (**Normal** case), or of a single, logarithmically transformed observation (**Log-Normal** case). If a scalar is specified, which must be the case in all but the **Two Samples** case, this value is the within treatment arm standard deviation common to all treatment arms. In the **Two Samples** case, a vector of length 2 specifies the standard variation on the treatment and control arms, in that order.

- **Binomial Proportion(s)), Binomial Odds, and Poisson Rate(s))** models

The variability of observations under these three probability models is governed by a mean-variance relationship.

However, designs are created using an approximation based on a single variance. The variability field is labelled **Variance Method**, which you can set to:

- `alternative` (the default), signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under the alternative hypothesis;
- `null`, signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under the null hypothesis;
- `intermediate`, signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under an intermediate hypothesis halfway between the null and alternative hypotheses;
- a scalar value specifying the within-group variance common to all treatment levels. In the **Two Samples** case, a vector of length 2 specifies variability on the treatment and control arms, in that order. This option is recommended for advanced users only.

In general, `null` will give designs with the most accurate size, and `alternative` the most accurate power. The `intermediate` option represents a compromise, and usually gives the most accurate sample size calculations. The asymptotic approximations used in power calculations tend to be more accurate under the **Binomial Proportion(s))** model than the **Binomial Odds** model.

- **Hazard Ratio** models

This field is absent, since the variability of the test statistic is approximately proportional to the number of events.

From the command line, set the variability using the `variance` argument to the `seqDesign` function.

Test Type

The **Test Type** field specifies the type of hypothesis test you're performing. Fixed sample testing usually discriminates among three hypotheses: the unknown mean is *consistent* with the null hypothesis ($H_0: \theta = \theta_0$); the unknown mean is *greater than* the null hypothesis ($H_+: \theta > \theta_0$); the unknown mean is *less than* the null hypothesis ($H_-: \theta < \theta_0$). Set the **Test Type** field to:

- two.sided for a two-sided hypothesis test. The two-sided test rejects H_+ and H_- , or H_+ and H_0 , or H_- and H_0 . The two-sided (lower) option is a special case required only when you are calculating the alternative hypothesis in a two-sided test. In this case, the two-sided (lower) option indicates that S+SEQTRIAL should calculate the alternative that the *lower* hypothesis test can detect, and the standard two.sided option indicates that S+SEQTRIAL should calculate the alternative that the *upper* hypothesis test can detect. If you are supplying the alternative hypothesis, and thus calculating either the sample size or the power, two.sided and two-sided (lower) are equivalent.
- greater for a one-sided hypothesis test against a greater alternative. Such a test does not distinguish between H_0 and H_- , and either rejects H_+ or H_0 .
- less for a one-sided hypothesis test against a lesser alternative. Such a test does not distinguish between H_0 and H_+ , and either rejects H_- or H_0 .
- equivalence for a one-sided equivalence test. This choice corresponds to a shifted one-sided hypothesis test that can be interpreted as testing for approximate equivalence in the sense that the decisions correspond to rejection of moderate superiority or moderate inferiority. Some statisticians refer to these as tests of "non-inferiority" and "non-superiority."

Note: In the statistical and clinical trials scientific literature, “equivalence tests” refer to designs which are actually just variants of one-sided (for non-inferiority or one-sided equivalence tests) or two-sided (for two-sided equivalence tests) hypothesis tests. In S+SEQTRIAL, “equivalence” is used to mean one-sided equivalence tests. When specifying a one-sided equivalence trial, you should enter the hypothesis of equality as the null hypothesis, and enter the hypothesis corresponding to the limit of acceptable inferiority as the alternative hypothesis. Such a design could also be obtained using “greater” or “less” hypothesis tests.

For example, with a two sample binomial proportions probability model, it may be the case that a new treatment would still be acceptably efficacious even if it truly had a response proportion that was 0.1 less than the response proportion with an existing efficacious treatment. When the level of significance and power are specified to obtain equal type I and II statistical errors, the exact same noninferiority design appropriate for this setting can be obtained by either

- specifying **Test Type** as **equivalence** with **Null proportions** set to **0.5, 0.5** and **Alt. proportions** set to **0.4, 0.5**, or
- specifying **Test Type** as **greater** with **Null proportions** set to **0.4, 0.5** and **Alt. proportions** set to **0.6, 0.5**.

In either case, the probability of observing a difference in response proportions above the critical value is less than the level of significance whenever the true response rate for the new treatment is 0.10 less than that for the existing efficacious treatment.

The default value is greater. If a hybrid test intermediate between a one-sided and two-sided test is defined using the **Epsilon** field on the **Advanced** tab, the **Test Type** field is automatically set to advanced. This field is disabled in the **k Samples** case.

From the command line, set the test type using the `test.type` argument to the `seqDesign` function.

Use Log Transform

By default, S+SEQTRIAL uses measures of treatment outcome and treatment effect that are more or less natural to non-statisticians. This is often counter to the way that the outcome is modeled statistically. For example, in logistic regression (the **Binomial Odds** model), the regression parameters are directly interpretable as the log odds ratio, rather than the odds ratio. Similarly, the regression parameters in the **Log-Normal Median(s)** model, the **Poisson Rate(s)** model, and the **Hazard Ratio** model refer to the log median ratio, the log rate ratio, and the log hazard ratio, respectively. Thus, many statisticians have become accustomed to dealing with logarithmic transformations of the natural parameters. You can choose to base input and output of hypotheses on the logarithmic scale by clicking on the **Use Log Transform** radio button.

From the command line, specify a log transform using the `log=T` argument to the `seqDesign` function.

Sequential Design

The **Sequential Design** groupbox allows you to specify interim analyses, and such additional parameters as the number and timing of the analyses and the stopping rules to apply. See Chapter 5 for a discussion of group sequential design aspects.

Save Design Object

Enter a name under which to save the results of the analysis in the **Save As** field. The name must be a valid S-PLUS object name—any combination of alphanumeric characters that starts with an alphabetic character is allowed. The only non-alphanumeric character allowed is the period (“.”). Names are case-sensitive. If an object with the name you enter already exists, its contents are overwritten. The saved object is of class `seqDesign`. If no name is supplied, the results are automatically saved under the name `last.seqDesign`.

SPECIFYING A GROUP SEQUENTIAL DESIGN

5

Overview	101
Group Sequential Design Concepts	102
Stopping Rules	102
Unifying Framework	105
Hybrid Tests	110
Boundary Scales	112
Partial Sum Scale	112
Sample Mean Scale	113
Z-Statistic Scale	114
Fixed-Sample P Scale	114
Error Spending Scale	115
Bayesian Scale	115
Conditional and Predictive Futility Scales	116
Boundary Shape	118
Group Sequential Design Families	118
Boundary Shape Functions	120
Unified Family (Sample Mean Boundary Scale)	123
Error Spending Function Family	126
Partial Sum and Normalized Z Statistic Families	129
Issues in Choosing a Stopping Rule	130
Primary Endpoint	130
Stopping Boundaries	130
Number and Timing of Interim Analyses	131
Basic Group Sequential Designs	132
Number of Analyses	132
Proportions of Sample Size (Sample Size groupbox)	133
Null Bnd. Shape	133
Alt Bnd. Shape	134

Advanced Group Sequential Designs	135
Operating Characteristics	136
Shape Parameters	137
Boundary	138
Boundary Matrices	138
Boundary Scale Parameters	140
Specifying Scale Parameters	140
Bayesian Scale Parameters	141
Futility Scale Parameters	146
Error Spending Scale Parameters	152
Constrained Boundaries	158
Types of Constraints	158
Specifying Constraints	159
Sample Mean Ordering	159
Arbitrary Stopping Rules	159

OVERVIEW

This chapter describes the design aspects of S+SEQTRIAL specific to group sequential tests, including:

- Important group sequential design concepts, such as the different types of early stopping rules and boundary scales, and the boundary shape parameterization used by S+SEQTRIAL;
- How to use the **Design** tab of the S+SEQTRIAL dialog to specify basic group sequential designs;
- How to use the **Advanced** tab of the S+SEQTRIAL dialog to achieve more complicated designs;
- Advanced group sequential design concepts, such as boundary scale parameters and arbitrarily constrained boundaries.

Group sequential designs are basically extensions to fixed sample designs, so this chapter assumes you are familiar with the contents of Chapter 4 on specifying fixed sample designs using S+SEQTRIAL.

GROUP SEQUENTIAL DESIGN CONCEPTS

The basic aspects of fixed sample design—specification of size, power, and sample size (see Chapter 4)—are all present in group sequential design. The difference is that with group sequential tests, sample size is no longer a single fixed number. Instead, the design focus for group sequential tests is selecting a stopping rule defining the outcomes that would lead to early termination of the study, along with an appropriate schedule for interim analyses. In this way, the average number of subjects exposed to inferior treatments can be decreased, and the ethical and efficiency considerations of clinical testing are better addressed.

Stopping Rules

In its most basic form, a group sequential design consists of a stopping rule for some statistic. At each interim analysis, a decision is made whether to stop the study early or to continue collecting data. The stopping rule takes the form of a sequence of critical values for some test statistic. Associated with each critical value is the decision to be made regarding the acceptance or rejection of the various statistical hypotheses.

When used as a statistical hypothesis test, a group sequential stopping rule is defined in the context of the statistical hypotheses (one-sided, two-sided, equivalence, or hybrid), the level of significance, and the power to detect some alternative hypothesis used in the design of the study. Therefore, all the issues of fixed sample hypothesis testing described in Chapter 4 apply to group sequential tests. In addition, however, specification of a stopping rule requires identification of:

- The number and timing of the interim analyses;
- The conditions under which early termination of the study would be considered—that is, whether the study should be stopped early for strong evidence of superiority, inferiority, approximate equivalence, or some combination of these three;
- The degree of conservatism to be used at the earliest interim analyses;
- The test statistic to be used in making any decision to terminate the study early.

Based on these choices, the critical values leading to early termination of the study are defined for each of the analyses in such a way that the operating characteristics of the hypothesis test (size and power, for example) are attained.

A number of families of group sequential stopping rules have been described in the statistical literature. Generally, these families have been grouped into four discrete classes according to the type of alternative hypothesis tested (one-sided or two-sided), and the conditions under which the study might be terminated early (decisions for the alternative, for the null, or both):

1. Single boundary designs for testing a one-sided hypothesis;

These designs only allow early stopping with a decision against the null hypothesis, or only with a decision not to reject the null hypothesis, but not both. Such designs are typical of those used in Phase II cancer clinical trials, where early stopping of the trial is generally indicated only if the drug does not look promising (that is, a decision not to reject the null hypothesis of no effect).

2. Two boundary designs for testing a one-sided hypothesis;

These designs allow early stopping for either a decision against the null hypothesis or a decision not to reject the null hypothesis. The triangular test of Whitehead & Stratton (1983), the one-sided symmetric tests of Emerson & Fleming (1989) and the one-sided tests of Pampallona & Tsiatis (1994) follow this general form. Shifting such tests to be centered about a treatment effect of zero has been used when attempting to establish approximate (one-sided) equivalence between two treatments based on demonstrating non-inferiority of the new treatment.

3. Two boundary designs for testing a two-sided hypothesis;

These designs allow early stopping for rejection of the null hypothesis, but a decision not to reject the null hypothesis can only occur at the final analysis. The Pocock (1977) and O'Brien & Fleming (1977) designs follow this general form, as do the Wang & Tsiatis (1987) generalizations of those designs and the Whitehead (1997) restricted procedure.

4. Four boundary designs for testing a two-sided hypothesis.

These designs allow early stopping for rejection of the null hypothesis or early stopping with a decision not to reject the null hypothesis. The Whitehead & Stratton (1983) double triangular test, the Emerson & Fleming (1989) two-sided symmetric tests, and the two-sided tests of Pampallona & Tsiatis (1994) follow this general form. These tests can be used to demonstrate approximate equivalence of two treatments in a two-sided fashion.

Figure 5.1 shows examples of the four types of stopping rules.

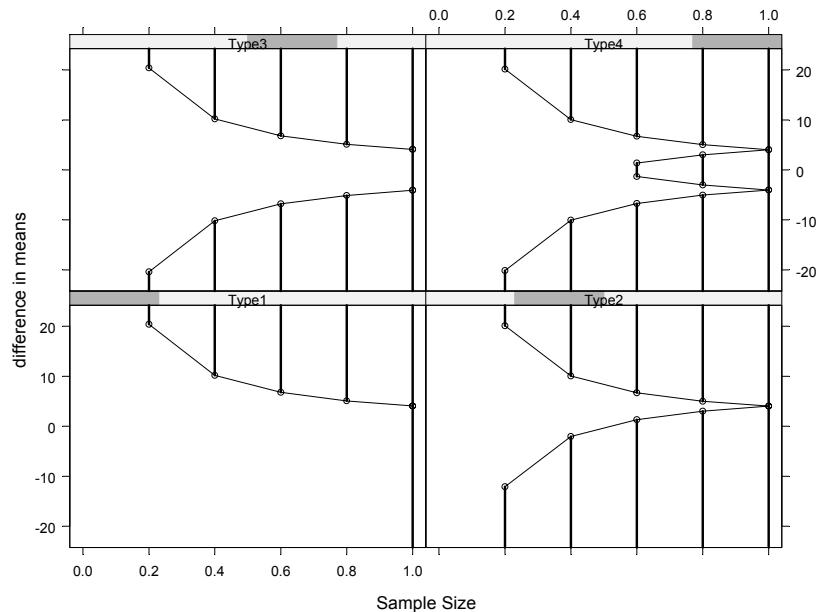


Figure 5.1: The four most common types of early stopping rules used in group sequential tests.

Figure 5.1 can be produced from the command line by typing:

```
> Type1 <- seqDesign(nbr.analyses=5, power=0.95,
+   test.type="greater",
+   early.stopping="alternative", P=1)
> Type2 <- update(Type1, early.stopping="both")
> Type3 <- update(Type1, test.type="two.sided",
+   early.stopping="alternative")
> Type4 <- update(Type3, early.stopping="both")
> seqPlotBoundary(Type1, Type2, Type3, Type4)
```

In these figures, the stopping rules are defined for the maximum likelihood estimate of the treatment effect. The horizontal axis represents the sample size available at each analysis, and the vertical axis represents the value of the maximum likelihood estimate of the treatment effect that might be computed at each sample size. The vertical lines represent the values of the test statistic that would lead to early termination of the study, and thus depict the stopping sets at each analysis. The decision to be made in each stopping region depends upon whether the hypothesis test is one-sided or two-sided, and in the case of one-sided tests, whether it is a one-sided test of a greater alternative or a lesser alternative.

The white space between the vertical lines depicts the values of the test statistic that would suggest continuation of the study, and thus those gaps represent the continuation sets. At the final analysis, the continuation sets are empty, so a single vertical line is drawn at that analysis.

The diagonal lines connecting the boundaries of the stopping regions are included only to allow better visualization of the relationships between stopping boundaries at successive analyses, and to identify the decision boundaries at the final analysis. These diagonal lines do not represent the stopping boundaries in a group sequential test.

Unifying Framework

S+SEQTRIAL implements the unifying framework of Kittelson & Emerson (1999) which treats the four discrete classes of stopping rules described above as special cases in a continuum of stopping rules. In this framework, a group sequential stopping rule can have up to four stopping boundaries. In S+SEQTRIAL, the lower boundary is labeled the “a” boundary, the lower inner boundary is labeled the “b” boundary, the upper inner boundary is labeled the “c” boundary, and the upper boundary is labeled the “d” boundary.

A study is terminated early if at any of the interim analyses the appropriate test statistic is:

- Lower than the “a” boundary;
- Between the “b” and “c” boundaries;
- Greater than the “d” boundary.

At any given interim analysis, values of the test statistic between the “a” and “b” boundaries or between the “c” and “d” boundaries correspond to a decision to continue the trial at least until the next analysis. In order to guarantee stopping the study at the final analysis, the “a” and “b” boundaries must meet at the final analysis, as must the “c” and “d” boundaries.

The decision of the hypothesis test that corresponds to each of the boundaries depends upon the type of alternative (for example, one-sided, two-sided, or equivalence) being tested:

- One-sided hypothesis tests of a greater hypothesis are often used in situations where beneficial effects of treatment are associated with larger values of the test statistic. In one-sided hypothesis tests of a greater alternative, test statistics less than the “a” boundary generally correspond to decisions rejecting the design alternative in favor of the hypothesis that the new treatment is not sufficiently beneficial to warrant adoption. Such decisions are often expressed as being for the null hypothesis. Test statistics exceeding the “d” boundary generally correspond to decisions rejecting the null hypothesis that the new treatment is ineffective or harmful in favor of the hypothesis that the new treatment has some beneficial effect, a decision that is often stated as being for the alternative hypothesis.
- One-sided hypothesis tests of a lesser hypothesis are often used in situations where beneficial effects of treatment are associated with smaller values of the test statistic (for example, when the outcome measures mortality rates). In one-sided hypothesis tests of a lesser alternative, test statistics less than the “a” boundary generally correspond to decisions rejecting the null hypothesis that the new treatment is ineffective or harmful in favor of the hypothesis that the new treatment has some beneficial effect in reducing the outcome measure. Such decisions are often expressed as being for the alternative

hypothesis. Test statistics exceeding the “d” boundary generally correspond to decisions rejecting design alternative in favor of the hypothesis that the new treatment is not sufficiently beneficial to warrant adoption, a decision that is often stated as being for the null hypothesis.

- One-sided hypothesis tests of equivalence (often termed non-inferiority tests) are shifted forms of the above one-sided tests. In these tests, the statistical hypothesis being tested is that the new treatment results in outcomes beyond some acceptable level of moderate inferiority. For instance, if beneficial outcomes were associated with smaller values of the test statistic, test statistics less than the “a” boundary correspond to decisions rejecting the hypothesis that the new treatment results in outcomes that are so much worse than the comparison group that it cannot be judged approximately equivalent. Such decisions are often expressed as being for the hypothesis of non-inferiority. In this setting, test statistics exceeding the “d” boundary generally correspond to decisions rejecting a hypothesis of moderate superiority. As a general rule, such a boundary is chosen to address the futility of continuing the study.
- Two-sided hypothesis tests are generally used when comparing two active treatments—when it is of scientific interest to demonstrate that either treatment is actually worse than the other. For instance, if situations in which treatment A is superior to treatment B are associated with larger values of the test statistic, then test statistics less than the “a” boundary correspond to decisions rejecting the null hypothesis of equivalent effects of the treatments in favor of the alternative that treatment B is superior to treatment A. Test statistics between the “b” and “c” boundaries correspond to decisions that neither treatment is markedly superior to the other (that is, a decision for the null hypothesis of approximate two-sided equivalence), and test statistics which exceed the “d” boundary correspond to decisions rejecting the null hypothesis of equivalent effects of the treatments in favor of the alternative that treatment A is superior to treatment B.
- Some authors also describe two-sided hypothesis tests of equivalence. However, from the discussion of the two-sided hypothesis tests it should be apparent that there is no

fundamental difference between a two-sided hypothesis test and a two-sided equivalence test. A properly powered two-sided hypothesis test allows any failure to reject the null hypothesis to be interpretable as rejection of unacceptable nonequivalence.

By appropriate definition of the four boundaries (“a,” “b,” “c,” and “d”), each of the four classes of previously described group sequential stopping rules (see page 102) can be obtained:

1. Single boundary designs for testing a one-sided hypothesis;

In single boundary designs, the “b” and “c” stopping boundaries are equal, so no early termination is possible for intermediate values of the test statistic. Depending on the conditions under which early termination of the study is desirable, either the “a” boundary is set to negative infinity or the “d” boundary is set to positive infinity at all interim analyses. The remaining single boundary is then set at some finite values that address the ethical and efficiency concerns of the study. Four variants of single boundary designs are of particular interest:

- One-sided tests of a greater alternative in which early termination of the study is desired only in the setting of strong evidence for the alternative. In this case, the “a” boundary is set to negative infinity, and the “d” boundary is used for early stopping.
- One-sided tests of a greater alternative in which early termination of the study is desired only in the setting of strong evidence against the alternative (and for the null). In this case, the “a” boundary is used for early stopping, and the “d” boundary is set to infinity.
- One-sided tests of a lesser alternative in which early termination of the study is desired only in the setting of strong evidence for the alternative. In this case, the “a” boundary is used for early stopping, and the “d” boundary is set to infinity.
- One-sided tests of a lesser alternative in which early termination of the study is desired only in the setting of strong evidence against the alternative (and for the null).

In this case, the “a” boundary is set to negative infinity at all interim analyses, and the “d” boundary is used for early stopping.

2. Two boundary designs for testing a one-sided hypothesis;

In one-sided hypothesis tests, the “b” and “c” stopping boundaries are equal at all analyses, so no early termination is possible for intermediate values of the test statistic. Both the “a” and “d” boundaries are used for early stopping, and these two boundaries meet at the final analysis.

3. Two boundary designs for testing a two-sided hypothesis;

In two-sided hypothesis tests with two early stopping boundaries, the “b” and “c” stopping boundaries are equal at all interim analyses, so no early termination is possible for intermediate values of the test statistic. Both the “a” and “d” boundaries are used for early stopping, but these two boundaries do not meet at the final analysis. At the final analysis, the “b” boundary is equal to the “a” boundary, and the “c” boundary is equal to the “d” boundary. Values of the test statistic between the “b” and “c” boundaries at the final analysis correspond to decisions for the null hypothesis (rejecting both the upper and lower alternative hypotheses).

4. Four boundary designs for testing a two-sided hypothesis.

In four boundary designs, the “b” and “c” boundaries differ at one or more interim analysis. Thus, these designs may allow early termination of the study for intermediate test statistic values at some of the interim analyses. It is often the case that the “b” and “c” boundaries are equal at the earliest interim analyses.

If at any interim analysis, the “a” boundary is at negative infinity, the “b” and “c” boundaries are equal, and the “d” boundary is at infinity, then all values of the test statistic correspond to continuation of the study. Such choices are tantamount to deciding no interim analysis takes place at that time. When this is true at all interim analyses, a fixed sample study results.

Hybrid Tests

Under the unifying framework of Kittelson & Emerson (1999), hybrid tests can be constructed which combine the features of some of the hypothesis tests described above. In this framework, every group sequential test can be viewed as the superposition of two one-sided hypothesis tests of the treatment effect θ :

- An *upper* hypothesis test of null hypothesis H_{0+} versus alternative hypothesis H_+ . The null hypothesis is that the treatment effect θ is less than or equal to some value θ_{0+} ($H_{0+}: \theta \leq \theta_{0+}$). The alternative hypothesis is that the treatment effect θ is greater than or equal to some value $\theta_+ > \theta_{0+}$ representing a minimal treatment effect that it is scientifically important to discriminate from the null hypothesis θ_{0+} ($H_+: \theta \geq \theta_+$).
- A *lower* hypothesis test of null hypothesis H_{0-} versus alternative hypothesis H_- . The null hypothesis is that the treatment effect θ is greater than or equal to some value θ_{0-} ($H_{0-}: \theta \geq \theta_{0-}$). The alternative hypothesis is that the treatment effect θ is less than or equal to some value $\theta_- < \theta_{0-}$ representing a minimal treatment effect that it is scientifically important to discriminate from the null hypothesis θ_{0-} ($H_-: \theta \leq \theta_-$).

In this framework, the “a” boundary rejects the null hypothesis H_{0-} of the lower hypothesis test, the “b” boundary rejects the alternative hypothesis H_- of the lower hypothesis test, the “c” boundary rejects the alternative hypothesis H_+ of the upper hypothesis test, and the “d” boundary rejects the null hypothesis H_{0+} of the upper hypothesis test.

The classical one-sided and two-sided hypothesis tests correspond to specifying particular relationships among the hypotheses of the upper and lower hypothesis tests of the unifying framework:

- One-sided tests of a greater hypothesis focus on the upper hypothesis test of the unifying framework. In this case, the lower hypothesis test is shifted up to coincide with the upper hypothesis test. The scientific hypotheses dictate the choices of the null hypothesis θ_{0+} and the alternative hypothesis θ_+ of the upper hypothesis test. The null and alternative hypotheses of the lower hypothesis test of the unifying framework are chosen to make the two tests coincide:
 $\theta_{0-} = \theta_+$ and $\theta_- = \theta_{0+}$.
- One-sided tests of a lesser hypothesis focus on the lower hypothesis test of the unifying framework. In this case, the upper hypothesis test is shifted down to coincide with the lower hypothesis test. The scientific hypotheses dictate the choices of the null hypothesis θ_{0-} and the alternative hypothesis θ_- of the lower hypothesis test. The null and alternative hypotheses of the upper hypothesis test of the unifying framework are chosen to make the two tests coincide: $\theta_{0+} = \theta_-$ and $\theta_+ = \theta_{0-}$.
- Two-sided tests use both the lower and upper hypothesis tests by ensuring that the null hypotheses for the lower and upper hypothesis tests coincide. The scientific hypotheses dictate the choices of the null hypotheses $\theta_{0-} = \theta_{0+} = \theta_0$ and the upper and lower alternative hypotheses θ_+ and θ_- .

When classical one-sided and two-sided hypothesis tests are desired, S+SEQTRIAL allows easy specification of the pertinent hypotheses in the usual fashion. As a general rule, the specification of the hypotheses of the lower and upper tests is controlled automatically through the specification of the type of hypothesis test desired, unless a special hybrid test is indicated. Even in those cases, the shift of the lower and upper hypothesis tests is controlled through two shift parameters, labeled ϵ in S+SEQTRIAL following the notation of Kittelson & Emerson (1999).

BOUNDARY SCALES

There has been some variation among the developers of group sequential methodology with respect to the notation used to specify stopping boundaries. In particular, different authors have defined their group sequential boundaries on different *boundary scales*, corresponding to different test statistics. This variation is most extreme between group sequential methods based on repeated significance testing, and those methods based on Bayesian approaches for monitoring clinical trials.

S+SEQTRIAL, however, makes no marked distinction among the various test statistics and boundary scales, since these scales can all be viewed as transformations of the test statistics routinely used when estimating or performing hypothesis tests. Furthermore, many of these scales can be viewed as quantifying various operating characteristics of particular stopping rules, so it is advantageous to blur the distinction between stopping rules defined for one test statistic (boundary scale) and those defined for other test statistics (boundary scales).

S+SEQTRIAL allows you to specify the desired boundary scale for all input to, and output from, the S+SEQTRIAL functions. Furthermore, group sequential design families are defined for several boundary scales, thus facilitating the selection of stopping rules that might have simple descriptions for certain test statistics (stochastic curtailment, for example). The interrelationships among the boundary scales allowed by S+SEQTRIAL are most easily described in the context of one sample tests of the mean of a normal distribution based on independent observations X_1, X_2, \dots, X_N . This setting serves as the underpinnings for all the computations performed in S+SEQTRIAL, and is easily generalized through asymptotic theory to all the probability models considered in this module.

Partial Sum Scale

For a null hypothesis of $H_0: \mu = \mu_0$ the Partial Sum scale corresponds to boundaries defined for

$$S_k = \sum_{i=1}^{N_k} X_i$$

In likelihood based inference, the efficient score function (before normalization using the information matrix) corresponds to this scale. This scale was used in the definition of the triangular and double triangular tests of Whitehead & Stratton (1983), and is the primary boundary scale used in the Whitehead (1997) text and the PEST3 statistical program (Brunier & Whitehead, 1993). This scale was also used in the definition of the one-sided and two-sided symmetric tests of Emerson & Fleming (1989).

This scale is the easiest one for computing the density of the group sequential test statistic. However, the partial sum statistic is not usually the most easily interpreted by clinical researchers. One exception is the case of a binary response. Here, the partial sum statistic measures the total number of events (for one arm trials) or the excess number of events in the treatment arm (for two arm trials).

Sample Mean Scale

For a null hypothesis of $H_0: \mu = \mu_0$, the Sample Mean scale corresponds to boundaries defined for

$$\bar{X}_k = \frac{S_k}{N_k}$$

The unified group sequential design family of Kittelson & Emerson (1999) is defined on this scale. The relationship between this scale and the Partial Sum scale should be obvious from the formula above.

Using the Sample Mean scale provides the stopping boundaries on a scale that is immediately interpretable in terms of the treatment effect. The test statistic on this scale corresponds to the maximum likelihood estimate of the treatment effect. Furthermore, for the **Normal Mean(s)** and **Binomial Proportion(s)** probability models, boundaries displayed on this scale are shape invariant under shift hypotheses, so this scale is best for graphical display of stopping boundaries. In S+SEQTRIAL, the Sample Mean scale is the default.

The Sample Mean scale refers to comparisons of the average response for **Normal Mean(s)** models, of the proportion of events for **Binomial Proportion(s)** models, of the odds of an event in the **Binomial Odds** models, of the event rate for **Poisson Rate(s)** models, and of the hazard ratio for **Hazard Ratio** models.

It should be noted that in some models, the natural parameter is actually modeled using a log transform. That is, in logistic regression

and survival analysis, the log odds ratio and log hazard ratio are actually modeled. The S+SEQTRIAL approach is to use the more natural parameter that would be understood by non-statisticians by default, so by default parameters are transformed back to the natural scale. However, if you prefer to use the log transformed natural parameter, S+SEQTRIAL allows the sample mean scale to also refer to the those log transformations. That is, by default S+SEQTRIAL considers the Sample Mean scale to apply to the odds ratio and hazard ratio, but this scale applies to the log odds ratio and log hazard ratio if you indicate that logarithmic transformations are to be used (see page 97).

Z-Statistic Scale

For a null hypothesis of $H_0: \mu = \mu_0$ the Z-Statistic scale corresponds to boundaries defined for

$$Z_k = \sqrt{N_k} \frac{(\bar{X}_k - \mu_0)}{\sigma}$$

There is an easy relationship between this scale and the Sample Mean scale. Through the use of this scale, statistics obtained from statistical packages (regression models, for example) can easily be compared to stopping boundaries. The tests described by Pocock (1977) were presented on this scale, and this scale is the primary boundary scale used in the text on group sequential methods by Jennison & Turnbull (1999).

Fixed-Sample P Scale

For a null hypothesis of $H_0: \mu = \mu_0$ the Fixed-Sample P scale corresponds to boundaries defined for the upper one-sided fixed sample p -value

$$P_k = 1 - \Phi(Z_k) = 1 - \int_{-\infty}^{Z_k} \frac{1}{\sqrt{2\pi}} \times e^{(-u)^2/2} du$$

Since the Fixed Sample P value is a straightforward transformation of the Z statistic, it is also a transformation of the sample mean scale.

This scale is sometimes useful when applying group sequential methods to situations where the test statistic is not normally distributed. Pocock (1977) found that in some such situations—for example, a test statistic with a t -distribution, or when analyzing

exponentially distributed data—approximately correct inference was obtained when the non-Gaussian test statistics were converted to their p -value, and then those fixed sample p -values were compared to the boundaries of normal theory-based group sequential tests expressed on the p -value scale.

Note: The Fixed Sample P values do not represent the p -value of the group sequential test statistic, nor do they have any set correspondence with a type I error spending function.

Error Spending Scale

In some group sequential designs it is of interest to consider the rate at which Type I or Type II error is allocated across analysis times. Such *error spending* functions are a very useful method of implementing group sequential stopping rules. The Error Spending scale can be an intuitively appealing scale on which to base stopping rules, though the efficiency of group sequential designs is not as well parameterized by the error spending function.

The exact formulas for the error spending function are notationally complex (see Emerson, 2000), but it easily shown that the error spending function is a transformation of the sample mean statistic.

Exact specification of a Error Spending scale is controlled by a number of boundary scale parameters described on page 152.

Bayesian Scale

The Bayesian properties of a particular stopping rule can be evaluated for a specified prior by considering the posterior probabilities of the various hypotheses. The Bayesian scale allows you to assess whether a decision made by a frequentist stopping rule is compatible with an inference that considers the prior probability that the treatment might be beneficial. Frequentist decisions are based solely on whether the observed data are commonly observed under specific hypotheses. Bayesian inference considers the more interesting question of whether a specific hypothesis is very probable given the observed data. By examining these Bayesian posterior probabilities under a variety of priors, you can assess whether a rejection of the null hypothesis is more likely to be a Type I error or a correct identification of the alternative hypothesis.

The exact formulas for the Bayesian posterior probabilities are notationally complex (see Emerson, 2000), but it easily shown that those posterior probabilities are transformations of the sample mean statistic. The transformations do, of course, involve the prior distribution that has been assumed for the treatment effect.

It should be noted that S+SEQTRIAL implements the “coarsened” Bayesian inference as described by Koprowicz, Emerson & Hoff (2002). In this approach, Bayesian posteriors are computed using a prior distribution for the treatment effect parameter θ and the approximate likelihood based on the large sample sampling distribution for the estimated parameter $\hat{\theta}$.

Exact specification of a Bayesian scale is controlled by a number of boundary scale parameters described on page 141.

Conditional and Predictive Futility Scales

In some cases, you may want to evaluate a stopping rule with respect to the probability that the decision made when stopping at some interim analysis might be different than the decision that would have been reached had the study not been terminated prematurely. When such criteria are used to terminate the study early, this is often referred to as *stochastic curtailment*. Evaluations of the stopping rule with respect to these criteria are based on the distribution of some test statistic at the final analysis conditional upon the value of the test statistic at an interim analysis. Of particular interest is the calculation of these probabilities under the assumption that the test statistic at the interim analysis falls exactly on one of the stopping boundaries for a group sequential design.

The conditional distribution of the test statistic at the final analysis is also dependent on the true treatment effect. Thus, in order to calculate the conditional probabilities, some assumption must be made about the true treatment effect. There are two general approaches used by S+SEQTRIAL in this regard:

- The Conditional Futility scale calculates the conditional probabilities by hypothesizing a particular value for the treatment effect. Major variants of this approach include using the null and alternative hypotheses from the hypothesis test, or using the current maximum likelihood estimate.

- The Predictive Futility scale calculates the conditional probabilities by assuming a prior distribution for the treatment effect, and then integrating the conditional probabilities calculated for each possible treatment effect over that prior distribution.

Either of these futility scales can be used to ascertain the probability that the decision made under early stopping might contradict the one that would have been made in the absence of interim analyses. As with the Error Spending and Bayesian scales, the formulas for these scales are notationally complex (see Emerson, 2000). However, all the variants of these futility scales are easily shown to be straightforward transformations of the sample mean scale.

Exact specification of the Conditional and Predictive Futility scales is controlled by a number of boundary scale parameters described on page 146.

BOUNDARY SHAPE

S+SEQTRIAL allows the specification of arbitrary stopping boundaries. (See page 158.) However, because the group sequential design for a clinical trial typically must satisfy certain constraints on the Type I and Type II errors, it is generally more useful to specify stopping boundaries based on design parameters that indicate the conditions under which early termination of the trial is possible, and the general “shape” of the stopping boundaries which allow early stopping.

Group Sequential Design Families

S+SEQTRIAL implements several large families of group sequential test designs. These include the most commonly used group sequential designs previously described in the statistical literature. Families are named according to a boundary scale of the test statistic that might be used in making a decision to terminate a clinical trial early.

As noted in the section on boundary scales (page 112), each scale is a straightforward transformation of the other scales, so the distinction between the families is not so much the stopping rules that may be specified, but the ease with which the relationship between stopping boundaries at the different analysis times can be described. A given stopping boundary may have a simple description on the basis of some of the boundary scales but not on others.

For example, the O’Brien-Fleming (1979) boundary relationships correspond to a constant threshold across analyses on the partial sum statistic scale and one of the conditional futility scales, and to a power function on the normalized Z statistic and sample mean scales. The O’Brien-Fleming relationships do not have a simple relationship on the error spending scale, however, as the error spending function for an O’Brien-Fleming design depends upon the size or power associated with that boundary, as well as the exact schedule of analyses.

Similarly, the Pocock (1977) boundary relationships correspond to a constant threshold across analyses on the normalized Z statistic and fixed sample P value boundary scales. Therefore they correspond to a power function on the partial sum and sample mean scales. They also do not have a simple relationship on the error spending scale,

because the error spending function for a Pocock design depends upon the size or power associated with that boundary and the exact schedule of analyses.

In general, it is often unimportant whether the boundary shape function is easily parameterized on a particular scale. There are some instances, however, that do make some parameterizations more useful than others. For example, because the average relative efficiency of Pocock (1977) boundaries under the alternative hypothesis is approximately the same under changes in the number and timing of analyses, and different choices of the size and power of the hypothesis tests, families which allow easy specification of such boundaries might be preferable to the use of a family based on error spending functions which do not have as constant behavior in this regard.

However, there are more issues involved in the selection of a group sequential test design than just efficiency based on average sample size. Thus, S+SEQTRIAL offers you flexibility in the choice of stopping rule specification. The families currently supported by S+SEQTRIAL include families based on the following scale:

- **Sample Mean scale**

This is the unified family of group sequential test designs described by Kittelson & Emerson (1999). This is the default family of group sequential designs, and as described below, includes the most commonly used group sequential designs as special cases.

- **Error Spending scale**

This family allows you to specify stopping boundaries according to a parameterization of the error spending functions for a group sequential design (Lan & DeMets, 1983; Pampallona, Tsiatis, & Kim, 1994).

- **Partial Sum scale**

This family describes boundary shapes for the partial sum statistic. This family has large overlap with the unified family of designs based on the sample mean scale, and so it is unlikely that you need to use this family.

- **Normalized Z Statistic scale**

This family describes boundary shapes for the normalized Z statistic. Again, this family has large overlap with the unified family of designs based on the sample mean scale, so it is unlikely that you need to use this family.

Boundary Shape Functions

S+SEQTRIAL implements families of group sequential designs that include the most commonly used group sequential designs that have been previously described in the statistical literature (for example, Pocock, 1977; O'Brien & Fleming, 1979; Whitehead & Stratton, 1983; Wang & Tsiatis, 1987; Emerson & Fleming, 1989; Pampallona & Tsiatis, 1994). Such designs differ with respect to the conditions under which early stopping is possible (see page 102), as well as differing in the degree of conservatism applied at the earliest analyses. The actual implementation of these designs is within the unified family of group sequential designs of Kittelson and Emerson (1999).

You can generally obtain these designs without understanding the full parameterization of the unified family of designs, because S+SEQTRIAL has a facility for obtaining such designs by specification of these restricted families directly. There are cases, however, when the clinical trial setting may require stopping rules that do not fall into any one of those previously described families. In such situations, you may need a more complete understanding of the parameterization of the boundary shape functions for each of the group sequential design families implemented in S+SEQTRIAL. This section contains a description of the boundary shape functions for those more advanced users desiring the full flexibility of the unified family of designs on the sample mean scale, or the family of designs based on the error spending function scale.

In each of the group sequential design families allowed by S+SEQTRIAL, the boundary relationships for the stopping boundaries at successive analyses are parameterized by a boundary shape function based on four parameters, denoted P, R, A, and G. In each case, the stopping boundary at a particular analysis is a function of the proportion of statistical information accrued so far, and the values of those four parameters. Letting Π represent the proportion of the planned maximal statistical information that is available at an

analysis, the stopping boundary for a boundary rejecting a null treatment effect of $\theta = 0$ in favor of a greater alternative would be to compare the appropriate test statistic (on a standardized scale) to

$$(A + \Pi^{-P} (1 - \Pi)^R) G$$

Typically, you specify P, R, and A when selecting a group sequential stopping rule, and S+SEQTRIAL finds a value of G which provides the desired size (Type I error) and power for the hypothesis test. The parameter R must be nonnegative, and constraints on the A and P parameters depend upon the choice of boundary scale for the group sequential design family as described below. However, it is possible to describe some general behavior of the boundary shape function.

At the beginning of the clinical trial $\Pi=0$, and at the final possible analysis $\Pi=1$. Considering those endpoints in the boundary shape function identifies the following interesting special cases:

- If $P > 0$, the boundary shape function approaches infinity at the beginning of the study, and it approaches AG at the final analysis. As P approaches infinity, the boundary shape function approaches infinity (for positive G) at all analyses prior to the final analysis.
- If $P = 0$ and $R = 0$, the boundary shape function is a constant $(A + 1)G$ throughout the study.
- If $P = 0$ and $R > 0$, the boundary shape function approaches $(A + 1)G$ at the beginning of the study, and AG at the final analysis. When $R = 1$, the stopping boundary is a straight line. When $R < 1$ the boundary tends to be concave downward, which tends toward more conservatism at the earliest analyses for typical choices of A and values of G. When $R > 1$ the boundary tends to be concave upward, which tends toward less conservatism at the earliest analyses for typical choices of A and values of G.
- If $P < 0$, then $R = 0$ is required. (Otherwise, a nonmonotonic boundary is obtained.) For $P < 0$ and $R = 0$, the boundary shape function approaches AG at the beginning of the study, and $(A + 1)G$ at the final possible analysis. When $P < -1$ the boundary tends to be concave downward, which tends toward more conservatism at the earliest analyses for typical choices of A and values of G. When $-1 < P < 0$ the boundary

tends to be concave upward, which tends toward less conservatism at the earliest analyses for typical choices of A and values of G.

All of the parameters can be thought of as relating to the conservatism of the decision to terminate the study at the earliest analyses. The way in which they affect that conservatism in terms of the shape of the stopping boundary is very different, however.

- P, when positive, is a measure of conservatism at the earliest analyses. The higher the value of P, the more difficult it is for a study to terminate at the earliest analyses. When P is infinite, the stopping boundary is infinite at all interim analyses.
- P, when negative, is also a measure of conservatism at the earliest analyses. The more negative the value of P, the more difficult it is for a study to terminate at the earliest analyses. Exactly how difficult it be to terminate at the earliest analysis relative to the final analysis be affected by the value of the A parameter. (It is difficult to compare the degree of conservatism for positive P and negative P, because the boundary shape is quite different.)
- R, when positive, is a measure of lack of conservatism at the earliest analyses. The higher the value of R, the less difficult it is for the study to terminate at the earliest analyses. The degree to which the value of R can affect the conservatism at the earliest analyses is greatly affected by the values of P and A. If P is also positive, the R parameter affects the curvature of the stopping boundary at the later analyses, but the P parameter has the greatest influence on the conservatism at the earliest analyses.
- A is a measure of separation between the first and last analyses, and thus can affect the conservatism of the test overall. When the G critical value is positive (as tends to be the case when P is positive or zero), a larger value of A tends to make the design less conservative at the earlier analyses. When the G critical value is negative, then A tends to be negative, and a more negative value of A makes the design less conservative at the earlier analyses. This behavior can be

deduced for some cases from the fact that when the magnitude of A is large, the difference between A and A+1 is less substantial.

Unified Family (Sample Mean Boundary Scale)

In the unified family of designs described by Kittelson and Emerson (1999), the stopping boundaries are ultimately defined for the maximum likelihood estimate of treatment effect. In this family, each of the four possible stopping boundaries (the “a,” “b,” “c,” and “d” boundaries) may have a distinct boundary shape function.

The full parameterization of the unified family of designs must also consider the size and power of the upper and lower hypothesis tests that were described in the unifying framework for group sequential stopping rules. For this reason, it is often difficult to provide exact ranges for the parameter values which result in a valid group sequential stopping rule: Some values of P, R, and A only results in valid group sequential stopping rules for selected values of the Type I or Type II errors. Thus, until you become familiar with the ranges of possible values, you may often encounter situations in which S+SEQTRIAL can not find a group sequential design satisfying the specified criteria. While it is possible that the problem is due to the starting estimates in the iterative search for a valid design, often it is due to the fact that no such design is possible.

It is also the case that many choices of P, R, and A, when used in combination with particular values for the size and power of the hypothesis test, result in group sequential designs which degenerate nearly to a fixed sample design. Such behavior can occur in two ways. In some cases, the boundary shape function is so flat on the sample mean scale that the probability is very nearly 1 that the trial would be stopped at the first interim analysis. In other cases, the boundary shape function is for all practical purposes infinite at the earliest analyses. In this case, the behavior of the group sequential design is much the same as if the number of analyses were reduced, omitting the earliest ones. The computational time can be quite lengthy with nearly degenerate designs, especially in the latter situation. Think carefully before asking for such extreme values of the unified family parameters.

The interpretation of the parameters P, R, and A are described below. Also included is some guidance about the ranges for the various parameters that include designs that are of the most practical interest.

P

When used alone (that is, when $R=0$ and $A=0$), a positive value for parameter P corresponds to the Wang & Tsiatis (1987) family of boundary relationships, also used by Emerson & Fleming (1989) and Pampallona & Tsiatis (1994). This family includes the boundary relationships of Pocock (1977) and O'Brien & Fleming (1979) as special cases. The parameterization used in S+SEQTRIAL differs slightly from those previous papers: the Pocock (1977) boundary relationship corresponds to $P = 0.5$, and the O'Brien-Fleming (1979) boundary relationship corresponds to $P = 1.0$. Higher values of P cause the group sequential test to be increasingly conservative at the earliest analyses, with an infinite value of P corresponding to a boundary that does not allow early termination of the study.

Special cases that might also be of interest when P is positive are:

- The triangular test's boundary relationship which corresponds to $P = 1$, $R = 0$, and $A = 1$;
- The family of sequential conditional probability ratio tests with boundary relationships having $P = 0.5$ and $R = 0.5$;
- An alternative parameterization of the O'Brien - Fleming boundary relationship which has $P = 1$, $R = 1$, and $A = 1$.

Boundaries with $P=0$ are prone to producing degenerate designs unless R is positive. However, it is possible to use a design with $P=0$ and $R=0$ for some (but not all) of the four boundaries in certain cases. As a general rule, the efficiency of such designs is very poor.

Boundaries with $P < 0$ must have $R = 0$. Furthermore, for typical choices of size (Type I error) and power, the value of A should be negative (see the discussion of the ranges of A below). As a matter of practicality, there is rarely much advantage in considering negative values of P much greater than -0.3 or much lower than -5 in this family.

As a practical rule, you should think carefully when using values for P much greater than 2.5 in the Wang & Tsiatis family of boundary relationships, especially if any of the interim analyses take place when less than 50% of the statistical information has accrued. Such designs have an extremely low probability of stopping at such early analyses for any of the hypothesized treatment effects that might be of practical interest. The algorithms used by S+SEQTRIAL in the iterative search for group sequential stopping rules can result in computational

times on the order of several minutes for such degenerate designs, due to the fact that the effective number of interim analyses is actually less than the nominal number.

Similarly, when negative values are used for P, careful thought should go into the use of P between 0 and -0.3, because such designs correspond to a very flat boundary shape function. Similarly, choices for P much less than -10 do not behave markedly different than those with P = -10. Furthermore, when large negative values are chosen for P along with a value of A close to -1, the design can degenerate very nearly to a fixed sample design in the sense that it is very difficult for the study to terminate early.

R

The parameter R cannot be negative. When R is positive and P = 0, the value of A must be positive for typical choices of the size (Type I error) and power.

In general, the efficiency of designs having nonzero R are not impressive. Some special designs described in the statistical literature do correspond to nonzero R, however. For example, the family of sequential conditional probability ratio tests have boundary relationships with P = 0.5 and R = 0.5, and an alternative parameterization of the O'Brien - Fleming boundary relationship has P = 1, R = 1, and A = 1.

There is little practical reason to consider designs with R less than 0.1 or R greater than 20, as that range of values encompasses a sufficiently broad spectrum of stopping rules.

A

As noted in the comments about the general properties of the boundary shape function, the A parameter's interpretation is probably best understood as the difference between the stopping boundary at the beginning of the study and the stopping boundary at the final analysis. The following rules of thumb were derived by considering that interpretation, along with the standardized scale on which the unified family of designs is defined, and typical choices for the size and power of hypothesis tests. The ranges of reasonable values of A are approximate, and were derived considering designs with two boundaries. Different behavior are obtained according to

the symmetry of the boundaries. In any case, the ranges given include many stopping rules that are not very useful, and the better designs tend to be in the interior of the valid ranges.

- When P is positive, but R = 0, A should be greater than -1. In order to avoid lengthy computations due to nearly degenerate designs, an approximate range for A is greater than 0.2 and less than 15 for designs having $0.1 < P < 2.5$.
- When P and R are both positive, A should be positive. In order to avoid lengthy computations due to nearly degenerate designs, an approximate range for A is greater than 0.2 and less than 15 for designs having $0.1 < P < 2.5$ and $0.1 < R < 20$.
- When P is 0 and R is positive, A should be positive. In order to avoid lengthy computations due to nearly degenerate designs, an approximate range for A is greater than 0.2 and less than 15 for designs having $0.1 < R < 20$.
- When P is negative and R = 0, A must be less than -1. In order to avoid lengthy computations due to nearly degenerate designs, an approximate range for A is less than -1.25 and greater than -15 for designs having $-10 < P < -0.1$.

Previously described group sequential boundary relationships with nonzero values for A include the triangular and double triangular tests of Whitehead & Stratton (1983), which correspond to choices of $P = 1$, $R = 0$, and $A = 1$, and an alternative parameterization of the O'Brien-Fleming boundary relationship which corresponds to choices of $P = 1$, $R = 1$, and $A = 1$.

G

Typically this parameter is not specified. Instead, it is calculated by S+SEQTRIAL.

Error Spending Function Family

The error spending function approach first described by Lan & DeMets (1983) has become a popular approach for implementing a particular stopping rule that allows flexible determination of the number and timing of interim analyses. Some authors have also suggested its use in the design of group sequential studies. In this approach, the stopping boundaries are determined to guarantee that a certain proportion of the total Type I error (or Type II error in the extension to these methods by Pampallona, Tsiatis, & Kim, 1995) is

“spent” at each of the analyses. The error spending function is typically defined by the cumulative Type I or Type II error spent by each analysis time.

As noted previously, the error spending function is just a transformation of the sample mean boundary scale, so there is a 1:1 correspondence between group sequential stopping rules expressed on one scale and group sequential stopping rules expressed on the other. The error spending function is, however, an extremely nonlinear transformation of the sample mean scale, and thus the scientific implications (as opposed to the statistical interpretation) of stopping boundaries expressed on the error spending scale are quite difficult to discern.

Furthermore, it is not widely recognized that the operating characteristics of a particular error spending function is not at all constant as the Type I or Type II error varies. For instance, in the various statistical papers written on the error spending approach (for example, Lan & DeMets, 1983, or Kim & DeMets, 1987), reference is made to error spending functions which approximate the O’Brien-Fleming (1979) or Pocock (1977) boundary relationships. In fact, no such error spending function is possible across all values of the size and power of a hypothesis test. There are different error spending functions for level .01, .025, .05, or .10 tests for each of these boundary relationships. In those papers, the error spending functions described tend to approximate the error spending functions of the one-sided level .025 O’Brien-Fleming or Pocock boundary relationships.

Of course, the dependence of the error spending functions for O’Brien-Fleming (1979) or Pocock (1977) on the size and power of the test is only important if your goal is to reproduce the behavior of those tests as you vary the size and power associated with the four stopping boundaries. Such is the case if, for example, you design say a level .05 two-sided test with 80% power to detect a particular design alternative. If for reasons of efficiency you desire to use Pocock (1977) boundary relationships (which tend to be very close to the stopping rule which averages the lowest sample size for treatment effects corresponding to the hypothesis being rejected by a given boundary), then you must use different error spending functions for the Type I error of .05 than you do for the Type II error of .20.

Similarly, the O'Brien-Fleming (1979) boundary relationships have an interpretation in terms of stochastic curtailment that is the same regardless of what size or power is used in the design of the group sequential test. The O'Brien-Fleming (1979) boundaries correspond to the decision to stop a study as soon as the conditional power of the test is less than 0.5. If the same behavior with respect to stochastic curtailment is desired for both the Type I and Type II errors, but those errors are not equal, then different error spending functions must be used to obtain the O'Brien-Fleming behavior.

Because of the nonintuitive, nonlinear scale used by the error spending function and the lack of generalizability of the operating characteristics mentioned above, the use of error spending functions in the design of a clinical trial is not recommended. It is much more straightforward to design a study on the sample mean scale, though the error spending approach remains an attractive approach for the flexible implementation of stopping rules.

Nevertheless, for those researchers who do desire to design studies on the error spending scale, S+SEQTRIAL implements a design family based on a generalization of the error spending function approach of Lan & DeMets (1983) and Pampallona, Tsiatis, & Kim (1995). In this family, you can set the error spending function for each of the four possible stopping boundaries (the “a,” “b,” “c,” and “d” boundaries) independently. The boundary shape function

$$(A + \Pi^{-P} (1 - \Pi)^R)G$$

is used to define the cumulative proportion of the Type I error (for the “a” and “d” boundaries) or Type II error (for the “b” and “c” boundaries) that is spent at the analysis in which proportion $0 < P < 1$ of the statistical information has been accrued. At the final analysis, it is assumed that all of the Type I and Type II error is to be spent, and thus all boundaries at the final analysis correspond to error spending functions of 1.

Boundaries on the error spending function range from 0 to 1. Because of this restricted range, boundary shape functions are only possible for certain combinations of the boundary shape function parameters:

- Negative values of P (with $R = 0$). In this setting, P measures the early conservatism of the stopping rule with more negative values of P corresponding to stopping rules that have lower probabilities of terminating the study at the earliest

analyses. Choosing a value of $P = -3.25$ approximates the operating characteristics of an O'Brien-Fleming boundary relationship for a one-sided Type I error of .025, although this error spending function based stopping rule does not exhibit the very extreme conservatism at the earliest analyses that is common with the O'Brien-Fleming boundary relationship.

- Positive values of R (with $P = 0$). In this setting, as R increases the stopping rule becomes less conservative at the earliest analyses.
- The interesting special case of $P = 0$ and $R = 0$ can be used to preclude early termination of the study.

In each case, the values of A and G are uniquely determined by the choice of P and R , so you never need specify either of these latter two parameters when using the error spending function family.

Partial Sum and Normalized Z Statistic Families

Boundary shape function families are also defined on the partial sum scale and normalized Z statistic scales. When $A=0$ and $R=0$, these families are related to the sample mean scale by an appropriate change in the value of the P parameter. A design with $P = p$ on the sample mean scale would correspond to a design with $P = p - 1$ on the partial sum scale, and $P = p - 0.5$ on the normalized Z statistic scale.

The stopping rules that correspond to these families with nonzero values for A or R generally differ from any in the unified family. The behavior of the A and R parameters are approximately the same in these families as it is in the sample mean family.

The boundary scales for these families are not particularly informative with respect to the scientific interpretation of early stopping, and they do not provide markedly different stopping rules than are available in the unified family. Therefore, it is unlikely that any particular advantage is gained through the use of these design families.

ISSUES IN CHOOSING A STOPPING RULE

Careful thought must be given to the conditions under which a clinical trial should be stopped prematurely. Early termination typically occurs because an experimental treatment is found to be more toxic than originally thought, or because the evidence suggesting one treatment is more efficacious than another is so strong that it would be convincing to the general medical community. In either case, the evidence in support of early termination of the study should be based on suitable statistical analyses. Such analyses are generally valid only if the stopping rules are specified in advance.

Primary Endpoint

When conducting a trial, you typically examine several clinical endpoints. These endpoints usually include different measures of toxicity, as well as different measures of efficacy. While it is important for all endpoints to be examined, analyses involving many endpoints tend to find more spurious “significant” results due to the multiple comparisons. Therefore, the usual practice in clinical trials is to identify the *primary endpoint* upon which the major statistical decisions are based. All other endpoints are then relegated to lesser roles in the decision to adopt a new treatment. (Secondary and tertiary outcomes may be used as supportive evidence for or against a new treatment, but are not considered sufficient on their own to show efficacy.) A stopping rule provides guidelines defining the primary outcome results that are judged so extreme as to warrant stopping the clinical trial prior to accruing the maximum sample size. Therefore, it is of utmost importance that the stopping rule be based on the outcome that provides the most evidence about the value of the new treatment.

Stopping Boundaries

It has become general practice in designing clinical trials that the efficacy boundary of the stopping rule be extremely conservative at the interim analyses. That is, the stopping rule should not usually suggest terminating a clinical trial early unless the results at the interim analysis have demonstrated not only a beneficial effect of the new treatment, but have actually demonstrated an effect well beyond the null hypothesis. This approach ensures that adequate information is gathered on safety and secondary endpoints, as well as ensuring that the results provide compelling evidence for the medical community.

Number and Timing of Interim Analyses

When you design a group sequential trial, you specify the total number of analyses that might be performed, as well as the sample sizes (or proportionate amount of statistical information) expected at each of those analyses. The statistical power of the test is somewhat affected by the exact choices for these two parameters. During the conduct of the clinical trial, however, departures from the planned number and timing of analyses pose no special problems for the validity of the statistical interpretation of the results, because there are methods that allow the flexible implementation of stopping rules so as to preserve the operating characteristics. Furthermore, so long as the departures from the planned number of interim analyses is slight, the effect of such departures on the statistical power of the test is minimal. (See Chapter 7 for information on how to modify stopping rules during a clinical trial.)

BASIC GROUP SEQUENTIAL DESIGNS

Basic group sequential designs in the unified family of designs (sample mean boundary scale) can be specified from the **Design** tab using the **Sequential Design** groupbox, as shown in Figure 5.2. More complicated group sequential designs can be specified from the **Advanced** tab, as described in the next section.

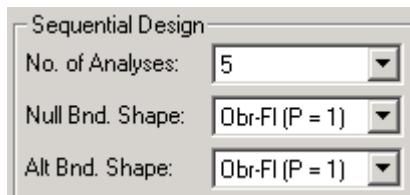


Figure 5.2: The **Sequential Design** groupbox on the **Design** tab.

Select a value greater than 1 in the **No. of Analyses** textbox to specify a group sequential design.

Note: Until you select a value greater than 1 in the **No. of Analyses** textbox, the other fields in the **Sequential Design** groupbox are disabled.

Number of Analyses

The hallmark of a group sequential test is that data are analyzed multiple times during the information accrual process. The necessary information includes the number and spacing of the interim analyses. It should be noted that in the actual conduct of a clinical trial, the actual number and timing of analyses may vary somewhat from the estimates used at the design stage. Using the methods described in Chapter 7, it is generally possible to allow such flexibility in the conduct of the clinical trial without sacrificing statistical rigor.

The **No. of Analyses** field specifies the number of planned analyses (interim and final) to be performed. By default the analyses are evenly spaced according to the sample size. For example, if the sample size is 100 and you specify 4 analyses, then analyses are performed after 25, 50, 75, and 100 measurements have accrued.

From the command line, specify the number of analyses using the `nbr.analyses` argument to the `seqDesign` function.

Proportions of Sample Size (Sample Size groupbox)

When a value greater than 1 is selected for the **No. of Analyses** field, the **Proportions of Sample Size: P_1, \dots** field in the **Sample Size** groupbox on the **Design** tab is enabled. When first enabled, this field displays the default value corresponding to equally spaced analyses. You may use this field to specify arbitrarily spaced analyses.

This field accepts a vector of length equal to the number of analyses specified in the **No. of Analyses** field. The values should be separated by commas, and the values must be strictly increasing in magnitude. If the final value is not 1, the vector is divided by the final value in order to obtain a vector of strictly increasing values ending with 1. Note that fractional representations are acceptable in order to avoid roundoff error.

From the command line, specify the timing of analyses using the `sample.size` argument to the `seqDesign` function.

Null Bnd. Shape

As discussed above (see page 118), S+SEQTRIAL uses four shape parameters to manipulate the boundary shape. The **Null Bnd. Shape** field on the **Design** tab allows you to specify the boundary relationships for the stopping boundary or boundaries associated with a decision for the null hypothesis (hence the boundary rejecting the alternative hypothesis) within the unified family of stopping rules defined for the sample mean scale. You may choose from selected values for shape parameter P as a scalar (in which case R=0 and A=0), specify the triangular test boundary shape function which corresponds to P=1, R=0, A=1, or specify that no trials should stop early with a decision for the null hypotheses (which corresponds to P=infinity, R=0, and A=0).

Alt Bnd. Shape As discussed above (see page 118), S+SEQTRIAL uses four shape parameters to manipulate the boundary shape. The **Alt Bnd. Shape** field on the **Design** tab allows you to specify the boundary relationships for the stopping boundary or boundaries associated with a decision for the alternative hypothesis (hence the boundary or boundaries rejecting the null hypothesis) within the unified family of stopping rules defined for the sample mean scale. You may choose from selected values for shape parameter P as a scalar (in which case R=0 and A=0), specify the triangular test boundary shape function which corresponds to P=1, R=0, A=1, or specify that no trials should stop early with a decision for the null hypotheses (which corresponds to P=infinity, R=0, and A=0).

Note: You can specify an arbitrary boundary shape vector $P = (P_\alpha, P_b, P_\sigma, P_d)$ on the **Advanced** tab, in which case the boundary shape fields on the **Design** tab is automatically set to Advanced. Similarly, you can specify values for the A and R parameters on the **Advanced** tab. The **Boundary Shape** field is also set to Advanced when a design is specified on the Error Spending scale.

From the command line, specify the boundary shape parameters using vectors for the P, R, and A arguments to the seqDesign function. The early.stopping argument to the seqDesign function can be used in conjunction with scalar valued boundary shape parameter arguments to obtain desired patterns of early stopping boundaries.

ADVANCED GROUP SEQUENTIAL DESIGNS

Advanced group sequential designs can be specified from the **Advanced** tab of the S+SEQTRIAL dialog, as shown in Figure 5.3.

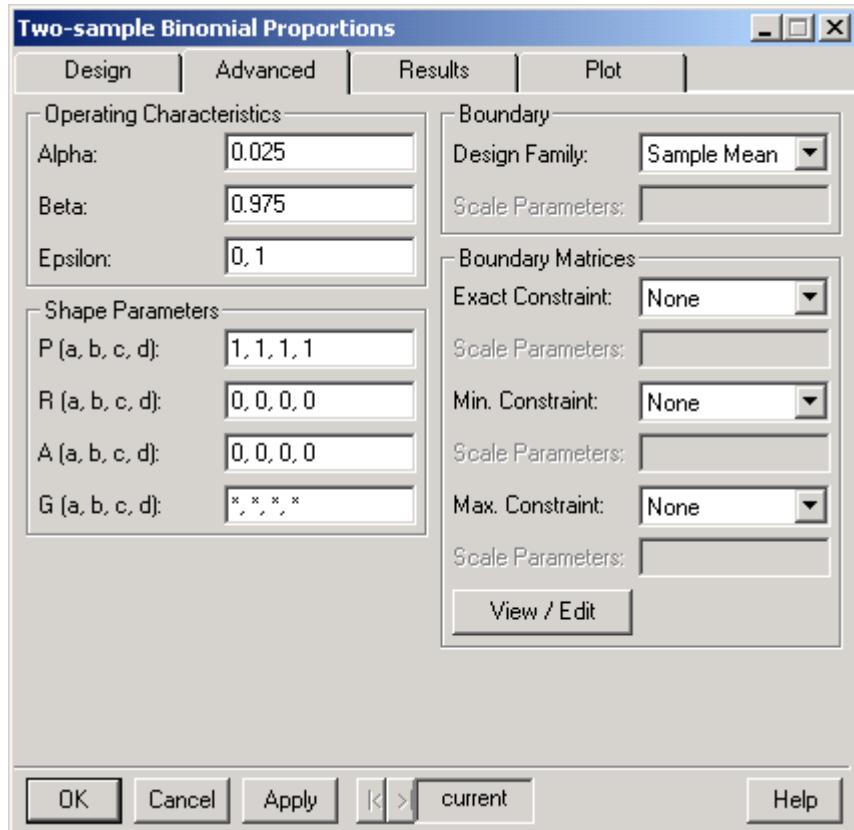


Figure 5.3: The **Advanced** tab of the **Two-sample Binomial Proportions** dialog.

Note: Until you select a value greater than 1 in the **No. of Analyses** textbox on the **Design** tab, all fields on the **Advanced** tab other than the **Operating Characteristics** are disabled.

On this tab, you can create designs using the full parameterization of the unifying framework of Kittelson & Emerson (1999), as well as specify designs in the other design families (for example, the error spending family). The meaning of the parameters corresponds exactly to that described in Kittelson & Emerson (1999).

Operating Characteristics

The **Operating Characteristics** groupbox gives you precise control over the test parameters.

Alpha

The **Alpha** field allows more advanced specification of the **Significance Level**. A vector of length two represents the one-sided level of significance for the lower hypothesis test and the one-sided level of significance for the upper hypothesis test. A scalar value specifies the level of significance of both the lower and upper hypothesis tests. The default value is half the value of the **Significance Level** for the two-sided test type, and otherwise equal to the value of the **Significance Level**.

If you change **Alpha**, the **Significance Level** field on the **Design** tab is automatically set to advanced.

From the command line, specify alpha using the `alpha` argument to the `seqDesign` function.

Beta

The **Beta** field allows more advanced specification of the upper and lower beta parameters. A vector of length two specifies beta for the lower hypothesis test and the upper hypothesis test. A scalar value represents the value of beta for both tests. The default value is one minus **Alpha**.

From the command line, specify beta using the `beta` argument to the `seqDesign` function.

Epsilon

The **Epsilon** field allows you to define a range of hybrid tests that form a continuum between one-sided and two-sided tests. A vector of length two represents the upward shift of the lower hypothesis test ε_l

and the downward shift of the upper hypothesis test ε_u . If both hypothesis shift parameters are equal to one (so the sum of those parameters is equal to two), the design corresponds to a two-sided test. If the sum of the hypothesis shift parameters equals one, the design corresponds to a one-sided test. A one-sided test tests against positive alternatives if $\varepsilon_l = 0$, and against negative alternatives if $\varepsilon_u = 0$. If $\varepsilon_l = \varepsilon_u = 0.5$, this corresponds to a test suitable for establishing one-sided equivalence (non-inferiority) between treatment results, providing the study is designed with sufficiently high power. See page 110 for more information on hybrid designs.

If you enter values in the **Epsilon** field, the **Test Type** field on the **Design** tab is automatically set to advanced. The **Epsilon** field is disabled in the **k Samples** case.

From the command line, specify epsilon using the `epsilon` argument to the `seqDesign` function.

Shape Parameters

As discussed above (see page 112), S+SEQTRIAL boundary shape functions are controlled by four parameters, denoted P, R, A, and G, each of which may be a vector of length 4. Such a parameterization is richer than is needed for most uses, but does allow the family to include all of the most commonly used group sequential designs.

In order to specify a boundary shape parameter X (one of P, R, A, or G), a vector of length 4 corresponds to (X_a, X_b, X_c, X_d) , describing each of the four stopping boundaries *a-d* (see page 102 for a discussion of the stopping boundaries).

Note: Any value(s) for P you enter on the **Advanced** tab override the setting on the **Design** tab.

From the command line, specify the boundary shape parameters using the `P`, `R`, `A`, and `G` arguments to the `seqDesign` function. These may be scalars or vectors of length 2 or 4, as described in the help file.

Boundary

As discussed above (see page 112), different authors define group sequential boundaries on different boundary scales. Basically these scales are just transformations of the test statistics routinely used when estimating or performing hypothesis tests about the mean of a distribution based on independent observations X_1, X_2, \dots, X_N .

Design Family

The **Design Family** field determines how the S+SEQTRIAL functions interpret boundaries. Select from among the supported boundary scales discussed on page 112.

From the command line, specify the boundary family using the `design.family` argument to the `seqDesign` function.

Scale Parameters

The **Scale Parameters** field allows you to enter additional parameters for the Error Spending, Bayesian, Conditional Futility, and Predictive Futility scales, and is disabled unless one of these scales is selected in the **Design Family** field. See page 140 for more information.

From the command line, specify the scale parameters as arguments to the `seqScale` function.

Boundary Matrices

The **Exact Constraint**, **Min Constraint**, and **Max Constraint** listboxes allow you to choose the scales for specifying the exact, minimum, and maximum constraints on the stopping boundary. (See page 158.) Select from among the supported boundary scales discussed on page 112.

Click on the **View/Edit** button to open a spreadsheet with columns for each of the four boundaries ($a - d$), and rows for each analysis time. Different boundary matrices are available for the exact, minimum, and maximum constraints. It is not necessary to save the spreadsheet to a file.

From the command line, specify the constraint matrices using the `exact.constraint`, `minimum.constraint`, and `maximum.constraint` arguments to the `seqDesign` function.

The **Scale Parameters** fields allow you to enter additional parameters for the Error Spending, Bayesian, Conditional Futility, and Predictive Futility scales. They are disabled unless one of these fields is selected in the corresponding **Constraint** listbox. See page 140 for more information.

From the command line, specify the scale parameters as arguments to the `seqScale` function.

BOUNDARY SCALE PARAMETERS

Complete specification of a Bayesian scale, Futility scale, or Error Spending Scale requires additional parameters.

Specifying Scale Parameters

Each boundary scale list box in the S+SEQTRIAL GUI is accompanied by a **Scale Parameters** field. This field is disabled unless the Bayesian, Conditional Futility, Predictive Futility, or Error Spending scale is selected in the associated boundary scale list box.

Scale parameters can be entered in any **Scale Parameters** field as a comma-delimited list of name=value pairs. For example, these are example parameters for the Bayesian scale:

```
priorTheta=0, priorVariation=10, threshold=0.40
```

From the command line, specify scale parameters as arguments to the seqScale function. For example, this command defines a Bayesian scale object called myScale:

```
> myScale <- seqScale("B", priorTheta=0,  
+ priorVariation=10, threshold=0.40)
```

Next, use the changeSeqScale function to change the appropriate scale in your design object. For example, given design object myDesign, this command would display boundaries on myScale:

```
> changeSeqScale(myDesign, outScale=myScale)
```

Boundaries can also be obtained on myScale by either of the following commands

```
> seqBoundary(myDesign, myScale)  
> update(myDesign, display.scale=myScale)
```

but since these commands recompute myDesign, they are somewhat slower than using changeSeqScale.

The specific parameters required by each scale are described below. For a more detailed treatment of these scales, refer to the *S+SEQTRIAL Technical Overview*. For the sake of brevity, examples are given using command line input.

Bayesian Scale Parameters

In a Bayesian setting, the treatment effect θ is assumed to be a random variable having some probability distribution. Before a clinical trial starts, you have limited knowledge about that probability distribution, but as you collect data in the clinical trial, your knowledge becomes more precise. Your initial knowledge about the probability distribution for θ is called the *prior distribution*. When you update that distribution by conditioning on the observed data, the new distribution is called the *posterior distribution* of θ .

Your decisions in this setting are based on probability statements about the hypotheses. That is, you might make decisions according to the probability that the null or alternative hypothesis is true. For a one-sided hypothesis test of a greater alternative— $H_0: \theta \leq \theta_0$ versus $H_+: \theta > \theta_+$, for example—you would be interested in one or more of the following statistics:

$$Pr(\theta \leq \theta_0 | \text{observed data}) \quad \text{probability that } H_0 \text{ is true}$$

$$Pr(\theta > \theta_0 | \text{observed data}) \quad \text{probability that } H_0 \text{ is false}$$

$$Pr(\theta < \theta_+ | \text{observed data}) \quad \text{probability that } H_+ \text{ is false}$$

$$Pr(\theta \geq \theta_+ | \text{observed data}) \quad \text{probability that } H_+ \text{ is true}$$

In order to calculate those statistics, you need to know:

- the probability model for the data for a particular value of θ ;
- the prior distribution of θ ;
- the value of the observed data.

Given these, S+SEQTRIAL can determine the cumulative distribution function (or equivalently the survivor function, which is just one minus the cumulative distribution function) for the posterior distribution of θ . From that posterior distribution, statistics similar to those given above can be computed.

Specifying the Probability Model

The probability model is specified when you create a design object, as described on page 73. S+SEQTRIAL considers probability models for which either the estimate of the treatment effect $\hat{\theta}$ or the logarithm of that estimate $\log(\hat{\theta})$ is approximately normally distributed. The latter is true for the multiplicative models (**Log-Normal Median(s)**, **Binomial Odds**, **Poisson Rate(s)**, and **Hazard Ratio**) when a log transform is not used (the default; see page 97). For all other probability models, it is the estimated treatment effect itself that is approximately normally distributed. It should be noted that S+SEQTRIAL implements the “coarsened” Bayesian inference as described by Koprowicz, Emerson & Hoff (2002). In this approach, Bayesian posteriors are computed using a prior distribution for the treatment effect parameter θ and the approximate likelihood based on the large sample sampling distribution for the estimated parameter $\hat{\theta}$.

Specifying the Prior Distribution

For those probability models having a normally distributed estimate of treatment effect, a convenient family of probability distributions to use for the specification of the prior distributions for θ is the normal distribution. This applies to the **Normal Mean(s)** and **Binomial Proportion(s)** probability models, as well as the multiplicative models when a log transform is specified. S+SEQTRIAL thus allows you to specify a prior distribution by providing the mean (priorTheta) and variance (priorVariation) of a normal distribution. A choice priorVariation = Inf yields a flat, or non-informative, prior distribution for the treatment effect θ .

For those probability models in which $\log(\hat{\theta})$ is approximately normally distributed (so $\hat{\theta}$ is approximately lognormal in distribution), a convenient family of probability distributions to use for the specification of the prior distributions for θ is the lognormal distribution. S+SEQTRIAL thus allows you to specify a prior distribution by providing the median (priorTheta) and shape parameter (priorVariation) of a lognormal distribution. It should be noted that because of the relationship between normal and lognormal distributions, you can also think of $\log(\text{priorTheta})$ as being the

mean and priorVariation as being the variance of a normal prior distribution for $\log(\theta)$. Thus, the choice priorVariation = Inf again corresponds to a non-informative prior on the log transformed scale. The probability models which use this definition of priorTheta and priorVariation are the multiplicative models (**Log-Normal Median(s), Binomial Odds, Poisson Rate(s), and Hazard Ratio**) when a log transform is not used (the default; see page 97).

Specifying the Critical Value

Once the prior distribution for θ has been specified, you need only indicate the hypothesis for which a posterior probability is desired. To increase the generality of the function, S+SEQTRIAL computes the posterior probability that θ exceeds some critical value (threshold). For example, if you specify a value of threshold corresponding to the null hypothesis θ_0 , S+SEQTRIAL returns the posterior probability that the true treatment effect exceeds θ_0 . This corresponds to the posterior probability that the null hypothesis is false. Similarly, if you specify a value of threshold corresponding to the alternative hypothesis θ_+ , S+SEQTRIAL returns the posterior probability that the true treatment effect exceeds θ_+ . This corresponds to the posterior probability that the alternative hypothesis is true.

Supplying Observed Values

The specification of Bayesian scale parameters described above can be used to cause S+SEQTRIAL to compute the posterior probability of arbitrary hypotheses (as indicated by the value of threshold) based on a particular observation of a group sequential test statistic. For example, suppose a seqDesign object named test.design contains information about a stopping rule in a normal probability model. Furthermore, suppose that at the third analysis, the estimate of treatment effect $\hat{\theta}$ is stored in an S+ variable theta.hat. To have S+SEQTRIAL compute the posterior probability of the null hypothesis (stored in object test.design), first define a Bayesian scale assuming, say, a prior distribution with mean 0 and variance 10:

```
> Bayes.scale <- seqScale("B", priorTheta=0, priorVariation=10,
+   threshold=test.design$hypothesis$theta.null)
```

After observing data corresponding to an estimated treatment effect of $\hat{\theta}$ at the third analysis, you can find the posterior probability of exceeding the null hypothesis by

```
> changeSeqScale(test.design, outScale=Bayes.scale,  
+     analysis.index=3, observed=theta.hat, inScale="X")
```

The posterior probability of other hypotheses can be found by varying the value of `threshold` in the definition of the Bayesian scale.

Unspecified Critical Value

When evaluating group sequential test designs, it is often of interest to consider the Bayesian posterior probability of the various hypotheses rejected by each of the boundaries. For instance, in a two-sided hypothesis test:

- The lower “a” boundary rejects the null hypothesis of the lower hypothesis test $H_{0-}: \theta \geq \theta_{0-}$;
- The inner “b” boundary rejects the lower alternative hypothesis $H_-: \theta \leq \theta_-$;
- The inner “c” boundary rejects the upper alternative hypothesis $H_+: \theta \geq \theta_+$;
- The upper “d” boundary rejects the null hypothesis of the upper hypothesis test $H_{0+}: \theta \leq \theta_{0+}$.

In judging the appropriateness of a particular stopping rule, it may therefore be of interest to consider at each stopping boundary the posterior probability that the hypothesis being rejected is false. That is, if you were to observe an estimated treatment effect at the first interim analysis of $\hat{\theta} = a_{\bar{x}1}$, it might be of interest to consider the posterior probability that $\theta < \theta_{0-}$, where that posterior probability is based on a specified prior distribution of θ and conditioned on the observation of $\hat{\theta} = a_{\bar{x}1}$. Similarly, if you were to observe an estimated treatment effect at the first interim analysis of $\hat{\theta} = c_{\bar{x}1}$, it might be of interest to consider the posterior probability that $\theta < \theta_+$.

This could be accomplished by considering four different Bayesian scales (one for each boundary) in which the value of threshold were chosen to correspond to the hypothesis being rejected.

S+SEQTRIAL simplifies this process. When a Bayesian scale is declared without specifying a particular value for threshold, it is assumed that you are only using that scale to display a stopping boundary, and that the desired effect is to compute the posterior probability that the hypothesis being rejected by a boundary is false. For example, the following commands can be used to display the relevant posterior probabilities for a stopping boundary:

```
> Bayes.scale <- seqScale("B", priorTheta=0, priorVariation=10)
> changeSeqScale(test.design, Bayes.scale)
```

Upon execution of this command, the stopping boundaries are displayed on a scale based on the prior distribution for θ in which

$$a_{Bj} = \Pr(\theta < \theta_{0-} | \hat{\theta}_j = a_{\bar{x}j})$$

$$b_{Bj} = \Pr(\theta > \theta_{-} | \hat{\theta}_j = b_{\bar{x}j})$$

$$c_{Bj} = \Pr(\theta < \theta_{+} | \hat{\theta}_j = c_{\bar{x}j})$$

$$d_{Bj} = \Pr(\theta > \theta_{0+} | \hat{\theta}_j = d_{\bar{x}j})$$

Pessimistic Priors

As described above, when you define a Bayesian scale without explicitly specifying a value for threshold, you are effectively using a different threshold for each of the four stopping boundaries. It is sometimes informative to carry this approach one step further, and use a different prior for each of the boundaries as well. This is usually done to incorporate a certain amount of conservatism into the priors used for each boundary. For instance, in order to behave conservatively when deciding to stop a study early with rejection of the null hypothesis (and acceptance of the alternative), you might want to consider a prior distribution that is centered on the null hypothesis. Similarly, when deciding to stop a study due to the futility of demonstrating a beneficial effect of the treatment (a rejection of the alternative and an acceptance of the null), you might want to use a prior distribution for θ that is centered on the alternative.

Such priors centered on the hypothesis opposite to the decision being made are sometimes termed *pessimistic priors*, and that terminology has been adopted in S+SEQTRIAL. Boundaries can be displayed using pessimistic priors by specifying a Bayesian scale using the `pessimism` parameter.

- Setting `pessimism` = 0 corresponds to using priors centered exactly on the hypothesis being rejected;
- Setting `pessimism` greater than 0 centers the priors for θ progressively further from the hypothesis being accepted by a stopping boundary (even further than the hypothesis being rejected);
- Setting `pessimism` less than 0 moves the center of the prior distribution closer to the hypothesis being accepted.

The `pessimism` parameter is measured in units of the standard deviation for the prior distribution.

Using this strategy, you can specify a Bayesian boundary scale by

```
> Bayes.scale <- seqScale("B",pessimism=0, priorVariation=10)
```

then display the stopping boundaries using

```
> changeSeqScale(test.design, Bayes.scale)
```

The “a” stopping boundary values are the posterior probability that the lower test’s null hypothesis is false, based on a prior distribution for θ that is centered on that null hypothesis θ_0 . The “b” stopping boundary values are the posterior probability that the lower test’s alternative hypothesis is false, based on a prior distribution that is centered on the alternative hypothesis θ_- . The “c” stopping boundary values are the posterior probability that the upper test’s alternative hypothesis is false, based on a prior distribution that is centered on the alternative hypothesis θ_+ . The “d” stopping boundary values are the posterior probability that the upper test’s null hypothesis is false based on a prior distribution that is centered on the null hypothesis θ_{0+} .

Futility Scale Parameters

One criterion that can be used to decide to stop a study early is whether the ultimate decision that would be made at the final analysis is already known with high confidence. For example, perhaps after

observing the results at an interim analysis, the probability that the null hypothesis ultimately be rejected is so low that further accrual of patients to the study is a waste of time and money. To use such a criterion, you must be able to calculate the probability that the test statistic at the planned final analysis would exceed some critical value. S+SEQTRIAL facilitates such an approach through the use of the Conditional and Predictive Futility scales.

Both of these scales are based on a probability distribution for the group sequential test statistic at the final analysis. That is, you might make decisions according to the probability that the results at the final analysis correspond to acceptance (or rejection) of the null or alternative hypotheses, with computation of those probabilities taking into account the value of the observed statistic at an interim analysis. For instance, in a two-sided hypothesis test of hypotheses $H_-: \theta \leq \theta_-,$ $H_0: \theta = \theta_0,$ and $H_+: \theta \geq \theta_+,$ your statistical decision rule might correspond to:

- Acceptance of H_- if the estimated treatment effect $\hat{\theta}_J$ at the final (J th) analysis is less than a critical value $a_{\bar{X}J};$
- Acceptance of H_+ if the estimated treatment effect $\hat{\theta}_J$ at the final (J th) analysis exceeds a critical value $d_{\bar{X}J};$
- Acceptance of H_0 if the estimated treatment effect $\hat{\theta}_J$ at the final (J th) analysis is between $a_{\bar{X}J}$ and $d_{\bar{X}J}.$

Therefore, at the j th interim analysis, given observation $\hat{\theta}_J = x,$ you might be interested one or more of the following statistics:

- The conditional probability that H_- will be accepted

$$Pr(\hat{\theta}_J \leq a_{\bar{X}J} \mid \hat{\theta}_J = x)$$

- The conditional probability that H_0 will be accepted

$$Pr(a_{\bar{X}J} < \hat{\theta}_J < d_{\bar{X}J} \mid \hat{\theta}_J = x)$$

- The conditional probability that H_+ will be accepted

$$Pr(d_{\bar{X}J} \leq \hat{\theta}_J | \hat{\theta}_J = x)$$

Statistics analogous to the above can all be computed knowing just the survivor distribution for the conditional distribution of $\hat{\theta}_J$ given an observed value of $\hat{\theta}_J$. S+SEQTRIAL facilitates these computations through the use of the Conditional and Predictive Futility scales for group sequential test statistics.

Specifying the Critical Value

When using the Conditional and Predictive Futility scales, specify a value for `threshold` parameter to indicate a particular critical value for $\hat{\theta}_J$ at the final analysis. S+SEQTRIAL computes the probability that the estimated treatment effect $\hat{\theta}_J$ would exceed the value of `threshold` at the J th analysis, conditional on the observation of the observed group sequential test statistic at the interim analysis.

Specifying the Treatment Effect

Conditional probabilities also depend upon the true value of the treatment effect θ , which is, of course, unknown. There are two basic strategies implemented in S+SEQTRIAL to deal with this problem:

- In the Conditional Futility (“C”) scale, supply a value of θ using the `hypTheta` parameter. This value is then used in the computation of the conditional probabilities.
- In the Predictive Futility (“H”) scale, supply a prior distribution for θ using the `priorTheta` and `priorVariation` parameters, or `pessimism` for transformations of stopping boundaries. (See the documentation for scale parameters for prior distributions under the Bayesian scale parameters on page 141.) The conditional probabilities are then found by integrating over the prior distribution.

For example, suppose `test.design` is a group sequential design object containing information about a stopping rule for a two-sided hypothesis test. Furthermore, suppose that at the third analysis, the estimate of the treatment effect $\hat{\theta}$ is stored in variable `theta.hat`. To compute the conditional probability that the observed treatment effect at the final analysis would correspond to acceptance of the

upper hypothesis H_+ , while taking the conservative approach of basing the computation on the assumption that the null hypothesis is true, first define a conditional scale:

```
> J <- length(sampleSize(test.design))
> crit.value <- seqBoundary(test.design, "X")[J,4]
> theta <- test.design$hypothesis$theta.null
> cond.scale <- seqScale("C", threshold=crit.value,
+   hypTheta=theta)
```

After observing data corresponding to an estimated treatment effect of $\hat{\theta}$ at the third analysis, find the conditional probability of accepting the upper alternative hypothesis by entering this command:

```
> changeSeqScale(test.design, outScale=cond.scale,
+   analysis.index=3, observed=theta.hat, inScale="X")
```

To compare that computation to an alternative strategy in which you use the current best estimate $\hat{\theta}_J$ as the true value of θ , you could use the following commands:

```
> cond.scale <- seqScale("C", threshold=crit.value,
+   hypTheta=theta.hat)
> changeSeqScale(test.design, outScale=cond.scale,
+   analysis.index=3,
+   observed=theta.hat, inScale="X")
```

To compute the conditional probability of accepting the upper alternative hypothesis using a prior distribution for θ , first define a predictive scale specifying that prior distribution. For example, specify a predictive scale assuming a prior distribution with mean 0 and variance 10 like this:

```
> pred.scale <- seqScale("H", priorTheta=0, priorVariation=10,
+   threshold=crit.value)
> changeSeqScale(test.design, outScale=pred.scale,
+   analysis.index=3,
+   observed=theta.hat, inScale="X")
```

A non-informative prior could have been used if `pred.scale` were defined using:

```
> pred.scale <- seqScale("H", priorTheta=0, priorVariation=Inf,
+   threshold=crit.value)
```

The conditional probability of other decisions can be found by varying the value of `threshold` in the definition of the scales.

Unspecified Critical Value

When evaluating group sequential test designs it is often of interest to consider the conditional probability that a decision would be made for the hypotheses rejected by each of the boundaries. For instance, in a two-sided hypothesis test:

- The lower (“a”) boundary rejects the null hypothesis of the lower hypothesis test $H_{0-}: \theta \geq \theta_{0-}$;
- The inner “b” boundary rejects the lower alternative hypothesis $H_-: \theta \leq \theta_-$;
- The inner “c” boundary rejects the upper alternative hypothesis $H_+: \theta \geq \theta_+$;
- The upper “d” boundary rejects the null hypothesis of the upper hypothesis test $H_{0+}: \theta \leq \theta_{0+}$.

In judging the appropriateness of a particular stopping rule, it may therefore be of interest to consider at each stopping boundary the conditional probability that the hypothesis being rejected at the interim analysis would not be rejected at the final analysis. For example, if you were to observe an estimated treatment effect at the first interim analysis of $\hat{\theta}_1 = a_{\bar{x}1}$, it might be of interest to consider the conditional probability that $\hat{\theta}_J > a_{\bar{x}J}$ at the J th (final) analysis, where that conditional probability accounts for the observation at the first interim analysis and is calculated using an assumed value for the true value of the treatment effect θ . Similarly, if you were to observe an estimated treatment effect at the first interim analysis of $\hat{\theta}_1 = c_{\bar{x}1}$, it might be of interest to consider the conditional probability that $\hat{\theta}_J > c_{\bar{x}J}$ at the J th (final) analysis. This could be accomplished by considering four different conditional scales (one for each boundary) in which the value of threshold were chosen to correspond to the hypothesis being rejected.

S+SEQTRIAL simplifies this process. When a conditional or predictive scale is declared without specifying a particular value for threshold, it is assumed that you are only using that scale to display a stopping boundary, and that the desired effect is to compute the conditional probability that the decision made at the interim analysis would have

been reversed at the final analysis. In the setting of transformation of entire stopping boundaries, there are several options for specifying the dependence of those conditional probabilities on the true treatment effect θ . Some of these options have the effect that in addition to using different thresholds for each of the different stopping boundaries, different values of θ are used as well.

For displaying stopping boundaries on conditional (“C”) scales that treat each of the four boundaries differently, you can choose to assume that the true value of θ corresponds to the either the hypothesis being rejected, or to the current maximum likelihood estimate (MLE) of the treatment effect.

- To assume that the true value of θ corresponds to the hypothesis being rejected, use the `hypTheta="design"` parameter. (This is the default value for the `hypTheta` argument.)
- To assume that the true value of θ corresponds to the current maximum likelihood estimate of the treatment effect, use the `hypTheta="estimate"` parameter.

For displaying stopping boundaries on predictive (“H”) scales that treat each of the four boundaries differently, you can choose to assume a specific prior distribution for θ , or to use separate pessimistic prior distributions for θ .

- To calculate conditional probabilities assuming a specific prior distribution for θ , use the `priorTheta` and `priorVariation` parameters. (See the documentation for Bayesian scale parameters on page 141 for the interpretation of `priorTheta` and `priorVariation` under the various probability models.)
- To calculate conditional probabilities using separate pessimistic prior distributions for θ , use the `pessimism` and `priorVariation` parameters. (Again, see the documentation for Bayesian scale parameters on page 141 for the interpretation of `pessimism` and `priorVariation` under the various probability models.) This approach uses a different prior distribution for each of the four stopping boundaries.

Error Spending Scale Parameters

Lan & DeMets (1983) described an approach to group sequential stopping rules that determines stopping boundaries according to the cumulative Type I error “spent” at, or prior to, a specific analysis. Pampallona, Tsiatis, & Kim (1994) extended that approach to Type II error spending functions. S+SEQTRIAL allows you to display group sequential test statistics and stopping boundaries as their error spending function equivalents. These error spending functions are easily shown to be 1:1 transformations of the other group sequential stopping statistic scales, so the S+SEQTRIAL implementation of error spending functions can be considered as additional scales for the display of group sequential test statistics.

It should be noted that there are some subtle nuances between the definition of the *scales* and the error spending *functions*. Error spending functions differ from error spending scales in that an error spending function does not include the probability that an observed treatment effect would be in the continuation sets for a given analysis. As a general rule, it is the error spending function that you want. The display of statistics on the more general error spending scales is only recommended for advanced users. For the sake of completeness, both are described here.

Specifying a Scale

The work of Lan & DeMets (1983) and Pampallona, Tsiatis, & Kim (1994) only considered one-sided hypothesis tests and two-sided hypothesis tests that were symmetric about the null hypothesis $\theta = \theta_0$. Thus, those authors only considered a single Type I error spending function (in the case of Lan & DeMets, 1983) or a single Type I error spending function and a single Type II error spending function (in the case of Pampallona, Tsiatis, & Kim, 1994).

S+SEQTRIAL allows a much richer family of designs in which each of the four boundaries may have distinct boundary shape functions, and the upper and lower hypothesis tests may be designed based on distinct choices of operating characteristics (level of significance and power). Furthermore, the upper and lower hypothesis tests may be arbitrarily shifted to obtain stopping rules that are in some sense intermediate between one-sided and two-sided hypothesis tests.

Therefore, S+SEQTRIAL implements four scales related to error spending functions for the display of a group sequential test statistic. The error spending scales are defined quite generally in order to

ensure a 1:1 correspondence between them and each of the scales for group sequential test statistics, and to allow their computation under arbitrary hypothesized values for the treatment effect θ . The error spending *functions* then correspond to a slight modification of the appropriate choice of one of the error spending *scales* evaluated at the stopping boundary for a particular choice for the hypothesized value of θ .

For an observed estimated treatment effect $\hat{\theta}_j = x$ at the j th analysis, the error spending scales can be defined as follows:

- The lower Type I error spending scale E_a ;

The probability that the estimated treatment effect at the j th analysis would be lower than x , plus the probabilities that the estimated treatment effect at any of the previously conducted interim analyses would be less than the corresponding “a” boundary, divided by the probability that the study would have stopped with an estimated treatment effect less than the “a” boundary.

- The lower Type II error spending scale E_b ;

The probability that the estimated treatment effect at the j th analysis would be greater than x , plus the probabilities that the estimated treatment effect at any of the previously conducted interim analyses would be between the corresponding “b” and “c” boundaries or greater than the corresponding “d” boundary, divided by the probability that the study would have stopped with an estimated treatment effect greater than the “b” boundary.

- The upper Type II error spending scale E_c ;

The probability that the estimated treatment effect at the j th analysis would be less than x , plus the probabilities that the estimated treatment effect at any of the previously conducted interim analyses would be between the corresponding “b” and “c” boundaries or less than the corresponding “a”

boundary, with that quantity divided by the probability that the study would have stopped with an estimated treatment effect less than the “c” boundary.

- The upper Type I error spending scale E_d .

The probability that the estimated treatment effect at the j th analysis would be greater than x , plus the probabilities that the estimated treatment effect at any of the previously conducted interim analyses would be greater than the corresponding “d” boundary, divided by the probability that the study would have stopped with an estimated treatment effect greater than the “d” boundary.

Select one of these scales using the `boundaryNumber` parameter to the `seqScale` function. The value of `boundaryNumber` may be one of “a”, “b”, “c”, or “d”.

Once the boundary number is chosen, you need only supply the hypothesized value of the treatment effect θ to be used in the computations. Specify this value using the `hypTheta` parameter. For example, suppose `test.design` is a `seqDesign` object containing information about a stopping rule in a normal probability model. Furthermore, suppose that at the third analysis, the estimate of treatment effect $\hat{\theta}$ is stored in a variable `theta.hat`. If you want to compute the value of the test statistic on, say, the lower Type I error spending scale (“a”) using a hypothesized value for θ equal to the null hypothesis (as stored in `test.design`), first define an error spending scale

```
> error.scale <- seqScale("E", boundaryNumber="a",
+   hypTheta=test.design$hypothesis$theta.null)
```

Then convert the group sequential test statistic to this error spending scale by entering

```
> changeSeqScale(test.design, outScale=error.scale,
+   analysis.index=3, observed=theta.hat, inScale="X")
```

Unspecified Scale

As a general rule, not all of the error spending scales are of interest for any given observed treatment effect. Furthermore, not every possible hypothesized value of θ is of interest with each of the error spending scales. Typically:

- The lower Type I error spending scale (“a”) is of most interest for observed values of the test statistic that are lower than the corresponding “a” boundary (even then, it is generally of interest only for a hypothesized value of θ that corresponds to the null hypothesis of the lower hypothesis test);
- The lower Type II error spending scale (“b”) is of most interest when evaluated for a hypothesized value of θ equal to the lower alternative hypothesis, and for values of the test statistic greater than the corresponding “b” boundary;
- The lower Type II error spending scale is of most interest when evaluated for a hypothesized value of θ equal to the upper alternative hypothesis, and for values of the test statistic less than the corresponding “c” boundary;
- The upper Type I error spending scale is of most interest when evaluated for a hypothesized value of θ equal to the null hypothesis of the upper hypothesis test, and for values of the test statistic greater than the corresponding “d” boundary.

To facilitate the evaluation of group sequential stopping rules, you can display the entire stopping rule on the error spending scales that make use of the above scheme. In this case:

- The “a” stopping boundary is displayed on the lower Type I error spending scale using the null hypothesis of the lower hypothesis test as θ ;
- The “b” stopping boundary is displayed on the lower Type II error spending scale using the lower alternative as θ ;
- The “c” stopping boundary is displayed on the upper Type II error spending scale using the upper alternative as θ ;
- The “d” stopping boundary is displayed on the upper Type I error spending scale using the null hypothesis of the upper hypothesis test as θ .

You can achieve this by defining an error spending scale in which neither the boundary nor the hypothesized value of θ are specified. For example, these commands

```
> error.scale <- seqScale("E")
> changeSeqScale(test.design, outScale=error.scale)
```

return the stopping boundaries on the error spending scales of greatest interest. Because no parameters are required for this combination of the error spending scales, it is sufficient to use the command

```
> changeSeqScale(test.design, outScale="E")
```

Error Spending Functions

The stopping boundaries displayed in the manner described above do not correspond exactly to the error spending *functions* which are probably of greater interest to you than the error spending *scales* would be. Error spending functions differ from the error spending scales in that the error spending function does not include the probability that an observed treatment effect would be in the continuation sets for a given analysis. In contrast, error spending scales for an observed estimated treatment effect $\hat{\theta}_j$ at the j th analysis might include some part of the probability that $a_{\bar{x}j} < \hat{\theta}_j b_{\bar{x}j}$ or $c_{\bar{x}j} < \hat{\theta}_j d_{\bar{x}j}$.

The error spending function corresponding to a particular group sequential stopping rule is returned by the function `seqDesign`, and stored in the `seqDesign` object. The error spending function can be retrieved from an existing `seqDesign` object using the function `seqBoundary`. For example, the error spending function for the group sequential design stored in `test.design` can be displayed using

```
> seqBoundary(test.design, scale="E")
```

Note that if you specify `display.scale="E"` in the call to `seqDesign`, the error spending function is returned, not the error spending scales.

In summary, S+SEQTRIAL defines four error spending scales which, while related to the error spending functions used by Lan & DeMets (1983) and Pampallona, Tsiatis, & Kim (1994), differ from those functions in important ways. If you want to make decisions about stopping a study based on statistics measured using the error spending concept, it is these error spending scales that would need to be used. The S+SEQTRIAL function `changeSeqScale` can be used to obtain the stopping boundaries or values of test statistics on one or more error spending scales.

However, you typically are most interested in error spending functions for the purposes of implementing a stopping rule so that the number and timing of interim analyses can be flexibly determined. It is the error spending function proper that is of interest here. Thus, for most purposes you merely want to display the error spending functions corresponding to a particular stopping boundary, and the S+SEQTRIAL function `seqBoundary` returns the error spending functions when argument `scale="E"`.

CONSTRAINED BOUNDARIES

Sometimes the stopping rule obtained from a parametric design family is unsatisfactory at one or more analyses. For instance, many clinical trialists find the extreme conservatism of the O'Brien-Fleming boundary relationships at the earliest analyses undesirable. One common modification of O'Brien-Fleming boundary relationships is to use the less extreme of the O'Brien-Fleming boundary, or a critical value corresponding to a fixed sample two-sided P value of .001. In order to facilitate this type of modification of stopping rules, S+SEQTRIAL allows you to specify constraints on the boundaries at particular analyses. All unconstrained boundaries are determined from a parametric design family in such a way as to maintain the desired operating characteristics (size and power) of the study design.

Types of Constraints

Constraints on the boundaries can be

- **Exact Constraints**

You enter the exact stopping boundary desired for a particular boundary ("a," "b," "c," or "d") at a specific analysis.

- **Minimum Constraints**

You enter a value for the stopping boundary that is the minimum value that you would like desired for a particular boundary ("a," "b," "c," or "d") at a specific analysis. If the parametric design family would result in a higher threshold for early termination at that analysis time, the boundary from the parametric family is used instead of this minimum constraint.

- **Maximum Constraints**

You enter a value for the stopping boundary that is the maximum value that you would like desired for a particular boundary ("a," "b," "c," or "d") at a specific analysis. If the parametric design family would result in a lower threshold for early termination at that analysis time, the boundary from the parametric family is used instead of this maximum constraint.

Specifying Constraints

Enter exact, minimum, and maximum constraints using the **Exact Constraint**, **Min Constraint**, and **Max Constraint** listboxes on the **Advanced** tab. Click on the **View/Edit** button to open a spreadsheet with columns for each of the four boundaries ($a - d$), and rows for each analysis time. Unconstrained boundaries must have NA in the corresponding place in the matrix.

From the command line, specify constraints as a `seqBoundary` object, or by naming a previous design with the same boundary.

If the group sequential design family is based on the sample mean, partial sum, or normalized Z statistic scales, the boundary constraints can be specified on any valid boundary scale *except* the error spending function scale. On the other hand, if the group sequential design family is based on the error spending scale, the boundary constraints can *only* be specified on the error spending function scale.

Sample Mean Ordering

When specifying the minimum or maximum constraints, the concepts of “minimum” and “maximum” are based on the ordering of the sample mean statistic. That is, one boundary is less than another if the boundary is lower on the sample mean scale. This distinction is important because some boundary scales have a reverse ordering.

For example, because the fixed sample P value scale is measured on the scale of a P value for a one-sided test of an upper alternative regardless of the type of hypothesis test being designed, a higher boundary on the sample mean scale actually corresponds to a lower number on the fixed sample P value scale. Thus, if you want to apply a constraint to avoid having the upper efficacy boundary of an O’Brien-Fleming test more extreme than the critical value of a fixed sample two-sided P value of .001, you would create a maximum constraint on the fixed sample P value scale that has .0005 in the appropriate position in the constraint matrix.

Arbitrary Stopping Rules

Through the use of exact constraints you can enter arbitrary stopping rules. When using the sample mean, partial sum, or normalized Z statistic design families, if the exact constraint matrix is fully specified (that is, there are no NA values in the exact constraint matrix), all group sequential design parameters are ignored except the alpha and beta parameters. S+SEQTRIAL returns a group sequential design

object having the specified stopping boundaries. The values of the alpha and beta parameters are used to find the hypotheses rejected by each boundary.

When an exact constraint matrix is fully specified on the error spending scale, S+SEQTRIAL returns a group sequential design having the specified error spending functions. In this way, arbitrary error spending functions can be used for group sequential test design.

EVALUATING DESIGNS

6

Overview	162
Stopping Boundaries	164
Boundary Scales	164
Tabulating Boundaries: Short Output	168
Tabulating Boundaries: Long Output	170
Plotting Boundaries	172
Inference on the Boundaries	175
Sample Size Distributions	176
Tabulating Sample Size Distributions	176
Plotting Sample Size Distributions	177
Power Functions	180
Tabulating Power Functions	181
Plotting Power Functions	182
Stopping Probabilities	185
Tabulating Stopping Probabilities	185
Plotting Stopping Probabilities	185
The Results Tab	187
Short Output	188
Long Output	189
Summary Tables	191
Options	194
Saved Results	194
The Plot Tab	195
Plot Types	196
Designs to Plot	197
Options	199
Reference Design	200

OVERVIEW

The process of selecting a group sequential design is usually an iterative one:

1. An initial design is specified;
2. The design is evaluated with respect to its operating characteristics;
3. The design is modified to obtain more desirable operating characteristics.

In a fixed sample study, where all data are accrued prior to any analysis, the operating characteristics of the test that tend to be examined most often are the size (Type I error), power (one minus the Type II error), and the sample size requirements. Additional operating characteristics that should perhaps be considered more often are the decision boundaries on a scale that has scientific (as opposed to merely statistical) meaning and, providing reasonable priors can be identified, the Bayesian interpretation of the decisions.

In the presence of a stopping rule, however, there are additional features of the design that might need to be examined. For example, the sample size accrued during the study is now a random variable, and hence summary statistics for that distribution might be of interest. This chapter covers:

- How to examine the stopping boundaries in both printed and plotted output;
- How to examine the sample size distribution for your design in both printed and plotted output;
- How to examine the power function for your design in both printed and plotted output;
- How to examine the stopping probabilities for your design in both printed and plotted output;
- How to use the **Results** tab of the S+SEQTRIAL dialog to print and save summary information about a design;
- How to use the **Plot** tab of the S+SEQTRIAL dialog to create evaluative plots.

Inference (p -values, point estimates, and 95% confidence intervals) is described in Chapter 8, Reporting Results.

STOPPING BOUNDARIES

S+SEQTRIAL computes stopping boundaries according to the design parameters you specify, and returns a design object containing the information related to those boundaries. A good first step in evaluating a design is to examine the stopping boundaries themselves.

Boundary Scales

As discussed in Chapter 5, different authors define their group sequential boundaries on different boundary scales. S+SEQTRIAL allows you to evaluate stopping boundaries on a number of scales. (See page 112 for a list of the supported scales.) Several of these scales are particularly useful for judging some of the operating characteristics of a stopping rule.

Sample Mean Scale

The sample mean scale expresses each boundary in terms of the maximum likelihood estimate (MLE) of the treatment effect. As noted in Chapter 8, the optimality properties of the maximum likelihood estimate are altered by the group sequential testing. The MLE tends to be more biased and have greater variability than several other estimators of treatment effect. Nonetheless, as a crude estimate of the treatment effect, the sample mean scale allows you to interpret the stopping boundaries on a scale that is more scientifically meaningful than other purely statistical scales.

When displaying the stopping boundaries graphically in S+SEQTRIAL, the sample mean scale is the default. This scale is invariant to hypothesis shifts in the additive probability models (normal and binomial proportions), so any symmetry in the treatment of the statistical hypotheses can be seen on this scale. This is not the case on the normalized Z statistic or partial sum scales. In the multiplicative models, if the hypotheses are treated symmetrically, the log link function causes some asymmetry in the display of stopping boundaries.

Examining the stopping rule on the sample mean scale may be of greatest interest to the data monitoring committee (DMC). Often members of the DMC have ethical concerns based on the crude estimate of the treatment effect (for example, that the trial should not be continued if the estimated hazard ratio in a survival study exceeds 3.0). Such criteria may be based on some unstated prior notion about

the likelihood that the new treatment is effective. It may be the case that the stopping rule must be chosen to reflect that prior notion. If the DMC is going to act on such criteria, it is important that the sampling distribution of the group sequential test statistic be computed according to the actual stopping rule.

Of course, the statistical precision of the crude estimates should also be taken into account before acting on any preconceived notion. Careful examination of the boundaries on the sample mean scale may serve to educate the DMC about the statistical evidence associated with those crude estimates at a particular analysis. For instance, by observing the stopping boundaries on the sample mean scale and considering the statistical properties of the stopping rule (such as the size and power), the DMC members may become convinced that their preconceived notions should be modified.

Conditional and Predictive Futility Scales

Clinical trialists are often concerned that early termination of a study might lead to a different decision than if the study continued to the planned final analysis. At other times, the concern is that the study might continue past a point that the ultimate decision is fairly certain. In this latter setting, the desire might be to terminate the study based on *stochastic curtailment* (see page 116). Such criteria are of course in 1:1 correspondence with boundaries defined according to other more scientifically (as opposed to purely statistically) interpretable scales, such as the sample mean scale. In fact, we generally find that when using stochastic curtailment as stopping criteria, clinical trialists generally define stopping rule parameters that are markedly (and at time inappropriately) inefficient. For that reason, we recommend that boundaries be defined on other scales and that futility concerns be examined by considering the changes in unconditional power introduced by a stopping rule relative to the power of a comparable fixed sample test.

Nevertheless, S+SEQTRIAL allows you to address questions related to the futility of continuing a study by considering the stopping boundaries on the conditional or predictive futility scales.

When displaying the stopping boundary on one of the futility scales, the stopping boundary at each analysis is interpretable as the conditional probability of reaching the opposite decision at the final analysis, conditioned on the observation of data that corresponds to the exact stopping boundary at the interim analysis. For instance,

when displayed on the futility scale, the “d” boundary at the first analysis corresponds to conditioning on an observation of data exactly on that boundary at the first analysis and computing the conditional probability that if the study continued, the data would correspond to an outcome below the “d” boundary at the final analysis.

Four variants of these scales are of particular interest (see the discussion beginning on page 146):

- Conditional futility scale with conditional probabilities for each boundary computed under the assumption that the true treatment effect is that defined by the hypothesis that is rejected by that boundary. This variant corresponds to computing at each analysis the conditional power of a reverse decision. This is the default conditional futility scale, and it is specified explicitly by specifying the boundary scale parameter `hypTheta = "design"`.
- Conditional futility scale with conditional probabilities for each boundary computed under the assumption that the true treatment effect is equal to the current maximum likelihood estimate at the stopping boundary for the interim analysis. This conditional scale is obtained by specifying the boundary scale parameter `hypTheta = "estimate"`.
- Predictive futility scale with conditional probabilities for each boundary computed by averaging over a Bayesian prior distribution for the treatment effect centered on the hypothesized treatment effect rejected by the boundary. This predictive scale is obtained by specifying the boundary scale parameter `pessimism = 0`, along with a value for the variability of the prior distribution.
- Predictive futility scale with conditional probabilities for each boundary computed by averaging over a non-informative Bayesian prior distribution for the treatment effect. This predictive scale is obtained by specifying the boundary scale parameter `priorVariation = Inf`, along with a value for the location of the prior distribution (using either `pessimism` or `priorTheta`).

Bayesian Scale

The frequentist inference based on *p*-values and confidence intervals is often criticized as being a precise answer to the wrong question. Frequentist inference quantifies the likelihood of observing the data obtained in a particular clinical trial under the assumption of particular hypotheses.

Bayesian inference attempts to answer a more appropriate question by computing the probability that a given hypothesis might be true conditional on the data obtained in the clinical trial. That computation is dependent upon the definition of a prior distribution for the treatment effect, and it is most often the case that the true prior distribution for the treatment effect is unknown. The result is that a Bayesian analysis is often criticized as being a vague answer to the right question.

In fact, both approaches have their uses. Every frequentist analysis should consider the Bayesian posterior probabilities of the hypotheses for a set of reasonable prior distributions, just as every Bayesian analysis should also consider the *p*-value associated with a particular outcome.

You can obtain the Bayesian posterior probabilities corresponding to a particular stopping rule by displaying the stopping boundaries on a Bayesian scale. S+SEQTRIAL allows you to define a prior distribution for the treatment effect based on the normal (for additive probability models) or lognormal (for multiplicative probability models) distribution.

Normalized Z statistic and Fixed Sample P value Scales

These boundary scales are statistically (as opposed to scientifically) focussed, and they are not as easily interpreted. Neither is accurate for inference in the presence of a group sequential stopping rule. Nevertheless, display of the stopping boundaries on these scales may be useful when actually monitoring the study, because many statistical programs return test statistics on this scale. Pocock (1977) has found that the use of stopping rules based on normally distributed data in settings where the test statistics are not truly normal is robust on the Fixed Sample P value scale.

Note that these considerations are unimportant if the S+SEQTRIAL function seqMonitor is used to monitor the trial. (See Chapter 7.) This function automatically computes the test statistics according to the probability model and makes the appropriate comparisons.

Tabulating Boundaries: Short Output

You can view the values for the stopping boundaries at each analysis as part of the standard printed output from S+SEQTRIAL (controlled by the **Short Output** option on the **Results** tab). These boundaries are reported using the current setting of the **Display Scale** field on the **Results** tab. The same information is also included in the **Summary Tables** output option on the **Results** tab.

For example, this is the printed output for the design tutorial.obf produced in Chapter 2.

```
STOPPING BOUNDARIES: Sample Mean scale
                     a      d
Time 1 (N= 77.83) -0.5000  0.3000
Time 2 (N= 155.67) -0.2500  0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000
```

From the command line, you can print the same information using the `print` function:

```
> print(tutorial.obf)
```

Or just type the object name:

```
> tutorial.obf
```

Note that these boundary values are reported on the Sample Mean scale (the default). You could view the conditional power corresponding to the stopping boundaries by choosing the Conditional Futility scale for the **Display Scale** on the **Results** tab.

```
STOPPING BOUNDARIES: Conditional Probability scale
(Conditional probability that estimated treatment effect
at the last analysis would correspond to an opposite
decision computed using hypothesized true treatment
equal to hypotheses being rejected)
                     a      d
Time 1 (N= 77.83) 0.5 0.5
Time 2 (N= 155.67) 0.5 0.5
Time 3 (N= 233.50) 0.5 0.5
Time 4 (N= 311.34) 0.5 0.5
```

```
Time 5 (N= 389.17) 0.5 0.5
```

The same output could be produced from the command line by any of the following commands

```
> changeSeqScale(tutorial.obf, seqScale("C"))
> seqBoundary(tutorial.obf, seqScale("C"))
> update(tutorial.obf, display.scale=seqScale("C"))
```

Of these methods, `changeSeqScale` has the least computational overhead.

The O'Brien-Fleming (1979) boundary relationships of design `tutorial.obf` correspond to stopping a trial early if the conditional power is below 50%.

It is perhaps at first surprising that the O'Brien-Fleming stopping rule, which is known to be conservative at the earliest analyses, would correspond to stopping a trial early for conditional power as high as 50%. This apparent paradox can be resolved by considering the conditional probability of a reverse decision at the final analysis on another of the conditional futility scales. Suppose that instead of assuming that the hypothesis being rejected is true (a rather bold assumption given the extreme conservatism of the O'Brien-Fleming boundary shape function), you assumed that the treatment effect is equal to the current best estimate. You could display the boundaries on this scale by entering `hypTheta = "estimate"` in the **Scale Parameters** field on the **Results** tab.

```
STOPPING BOUNDARIES: Conditional Probability scale
  (Conditional probability that estimated treatment effect
   at the last analysis would correspond to an opposite
   decision computed using hypothesized true treatment
   equal to maximum likelihood estimate)
      a      d
Time 1 (N= 77.83) 0.0000 0.0000
Time 2 (N= 155.67) 0.0000 0.0000
Time 3 (N= 233.50) 0.0169 0.0169
Time 4 (N= 311.34) 0.1302 0.1302
Time 5 (N= 389.17) 0.5000 0.5000
```

The same output can be produced from the command line by typing

```
> changeSeqScale(tutorial.obf,
+     seqScale("C",hypTheta="estimate"))
```

At the earliest analyses, the conditional probability of a reverse decision at the final analysis is exceedingly small.

You might also consider the Bayesian inference that would correspond to the stopping rule for design tutorial.obf by displaying the boundaries on a Bayesian scale. To consider a non-informative prior distribution for the treatment effect, you can display the boundaries on this scale by choosing the Bayesian scale for the **Display Scale** on the **Results** tab, and entering `priorTheta = 0, priorVariation=Inf` in the **Scale Parameters** box.

From the command line, type

```
> changeSeqScale(tutorial.obf,  
+     seqScale("B",priorTheta=0,priorVariation=Inf))
```

Tabulating Boundaries: Long Output

You can obtain more detailed printed output about the stopping boundaries using the **Long Output** option on the **Results** tab. For example, this is part of the **Long Output** for the design tutorial.obf.

HYPOTHESES:

Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:

```
Null hypothesis : Theta >= 0      (size   =  0.025 )  
Alternative hypothesis : Theta <= -0.2    (power  =  0.975 )
```

Boundary hypotheses:

```
Boundary a rejects : Theta >=  0.0      (lower size   =  0.025 )  
Boundary b rejects : Theta <= -0.2      (lower power  =  0.975 )  
Boundary c rejects : Theta >=  0.0      (upper power  =  0.975 )  
Boundary d rejects : Theta <= -0.2      (upper size   =  0.025 )
```

FULL PARAMETERIZATION: (Emerson & Fleming (1989) symmetric test)

Number and timing of analyses:

	Time 1	Time 2	Time 3	Time 4	Time 5
Combined sample size	77.83	155.67	233.5	311.3	389.2
(Treatment arm)	38.92	77.83	116.8	155.7	194.6
(Comparison arm)	38.92	77.83	116.8	155.7	194.6
Cum proportion	0.20	0.40	0.6	0.8	1.0

Size, power, and hypothesis shift parameters:

	Lower	Upper
Alpha	0.025	0.025
Beta	0.975	0.975

```

Epsilon 1      0

Alpha      (size = 0.025)
Beta       (type I and type II errors equal)
Epsilon     (one-sided test of lesser alternative)

Design family based on: Sample Mean scale

Boundary shape parameters:
  P A R      G
  a 1 0 0 2.0134    (O'Brien-Fleming)
  b 1 0 0 2.0134    (O'Brien-Fleming)
  c 1 0 0 2.0134    (O'Brien-Fleming)
  d 1 0 0 2.0134    (O'Brien-Fleming)

Constraints -
  Exact constraints: (none)
  Minimum constraints: (none)
  Maximum constraints: (none)

STOPPING BOUNDARIES: Sample Mean scale
                  a      d
Time 1 (N= 77.83) -0.5000  0.3000
Time 2 (N= 155.67) -0.2500  0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000

STOPPING BOUNDARIES: Error Spending Function scale
                  a      d
Time 1 (N= 77.83) 0.0001  0.0001
Time 2 (N= 155.67) 0.0292  0.0292
Time 3 (N= 233.50) 0.1974  0.1974
Time 4 (N= 311.34) 0.5542  0.5542
Time 5 (N= 389.17) 1.0000  1.0000

STOPPING BOUNDARIES: Standardized Cumulative Sum scale
                  a      d
Time 1 (N= 0.2) -2.0134  1.2081
Time 2 (N= 0.4) -2.0134  0.4027
Time 3 (N= 0.6) -2.0134 -0.4027
Time 4 (N= 0.8) -2.0134 -1.2081
Time 5 (N= 1.0) -2.0134 -2.0134

```

From the command line, you can print the same information using the `details=T` argument to the `print` function:

```
> print(tutorial.obf, details=T)
```

Plotting Boundaries

You can plot the stopping boundaries using the **Decision Boundaries** option on the **Plot** tab. The boundaries are plotted using the current setting of the **Display Scale** field on the **Plot** tab.

For example, Figure 6.1 shows a plot of the stopping boundaries for the design `tutorial.obf`. The *x*-axis shows that the interim analyses are planned after 78, 156, 234, and 312 subjects are accrued, and the final analysis after 390 subjects; the *y*-axis shows the stopping boundaries on the sample mean scale.

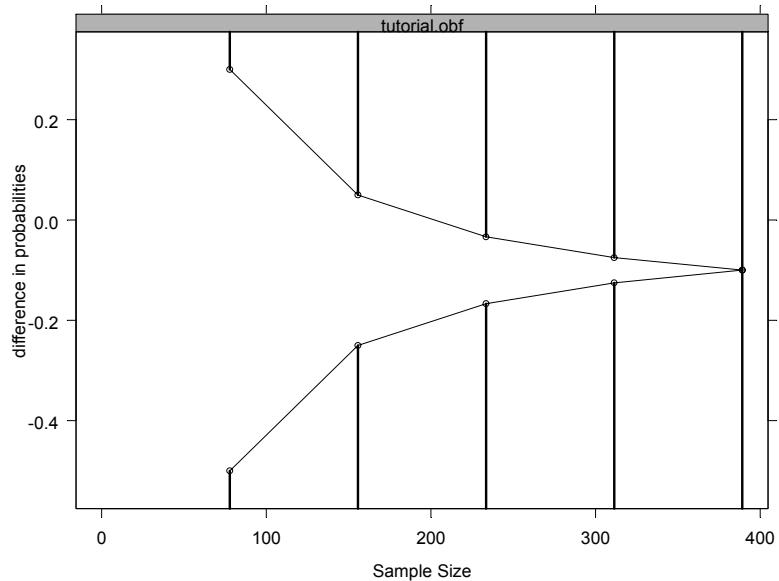


Figure 6.1: Plot of the stopping boundaries for design *tutorial.obf*.

From the command line, you can plot stopping boundaries using the `seqPlotBoundary` function. For example:

```
> seqPlotBoundary(tutorial.obf, fixed=F)
```

You can use the **Matching Fixed Designs** option in the **Designs to Plot** groupbox to plot the stopping boundary for a fixed design with the same power. For example, Figure 6.2 shows a plot of the stopping boundaries for the design *tutorial.obf* produced in Chapter 2, and for a fixed design matched for power (compare Figure 6.1). The fixed design always stops after 370 subjects have been accrued.

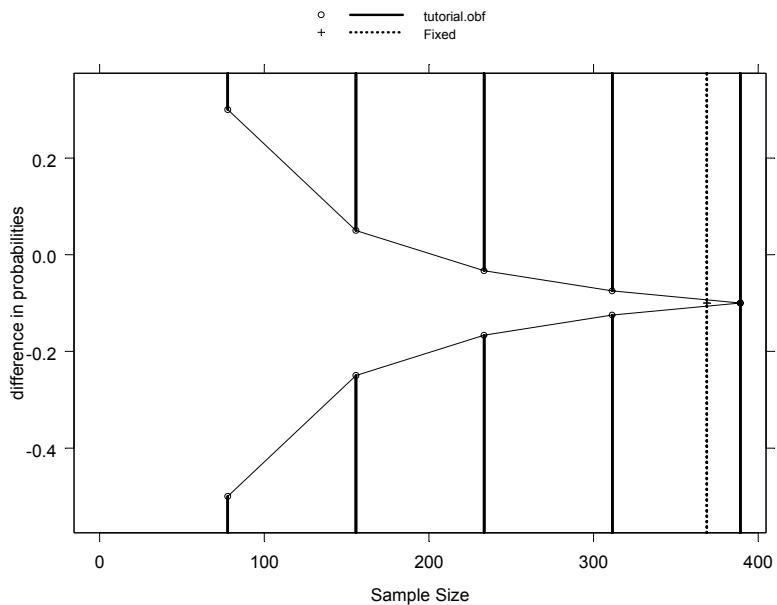


Figure 6.2: Plot of the stopping boundaries for design `tutorial.obf`, and for a fixed design with the same power (compare Figure 6.1).

From the command line, you can plot a matching fixed design using the `fixed=T` argument to the `seqPlotBoundary` function. For example:

```
> seqPlotBoundary(tutorial.obf, fixed=T)
```

Note that the `fixed` argument defaults to `TRUE` if only one `seqDesign` object is supplied to the `seqPlotBoundary` function, so the `fixed` argument could have been omitted in this case.

INFERENCE ON THE BOUNDARIES

In assessing whether a particular stopping rule is appropriate for the clinical trial setting at hand, it is important to consider the statistical inferences to be made when the study is over. This is equally important at the interim and final analyses. The issues that need to be examined are:

- *With what confidence can a decision be made if the study is terminated early?*

This is addressed by considering the adjusted p -value (see Chapter 8 on page 235) that would correspond to a decision to stop the study.

- *What estimated treatment effects correspond to decisions for the null and alternative hypotheses?*

This is addressed by considering the adjusted estimates of treatment effect (see Chapter 8 on page 231). It can sometimes happen that the decision to reject the null hypothesis corresponds to an estimated treatment effect that is not judged clinically important. In such a situation, the study may be overpowered. It can also happen that a decision not to reject the null hypothesis corresponds to an estimated treatment effect that many in the scientific community would regard as a clinically important improvement. In this latter setting, the study may be underpowered.

- *What will be the precision of the estimate of treatment effect?*

This is addressed by considering the adjusted confidence intervals (see Chapter 8 on page 233). The confidence intervals can be viewed as the set of hypothesized treatment effects that have not been rejected by the clinical trial results. If the confidence intervals do not discriminate between hypotheses that are scientifically important to distinguish (for example, the confidence interval includes both the null and alternative hypotheses), the study may be underpowered.

A full discussion of proper inference following a group sequential test is included in Chapter 8.

SAMPLE SIZE DISTRIBUTIONS

In addition to examining the maximal sample size required by the design, it is important to remember that in a group sequential setting the actual sample size accrued when the study is terminated is a random variable. Thus, you can characterize the operating characteristics of a test with respect to the distribution of sample sizes at the time of study termination. Often this distribution is characterized by the expected number of subjects accrued prior to study termination, the average sample number (ASN), although other summary measures of the sample size distribution (the 75th percentile, for example) are appropriate in specific situations. Two tests with the same level of significance and the same statistical power to detect a particular alternative may have very different probability distributions for the sample size at the time the study is terminated. In general, the sample size distribution is a function of the stopping boundaries and the value of the true mean.

In effect, the extent to which a particular group sequential design addresses the ethical or efficiency concerns that caused a researcher to consider interim analyses as the data accrued is a direct function of the number of subjects typically required by the design when the true treatment effect corresponds to lack of benefit (or extreme harm) or corresponds to extreme benefit. A “good” group sequential design would be one in which the mean (or median or 75th percentile, etc.) of the sample size distribution was suitably low when the true treatment effect would suggest it is unethical (or inefficient) to conduct a clinical trial.

Tabulating Sample Size Distributions

You can tabulate the average sample number for five (by default) values of the treatment effect (theta) using the **Summary Tables** option on the **Results** tab. For example, here are the values for the `design tutorial.obf` produced in Chapter 2.

Operating characteristics		
Theta	ASN	Power.lower
-0.20	241.1993	0.9750
-0.15	286.3289	0.8371
-0.10	308.6648	0.5000
-0.05	286.3294	0.1629
0.00	241.1998	0.0250

From the command line, you can print the same information using the `summary` function (which uses `summary.seqDesign`). For example:

```
> summary(tutorial.obf)
```

You can tabulate the average sample number for any number of hypothesized values of the treatment effect using the **Saved Results** groupbox on the **Results** tab. You can also tabulate arbitrary percentiles. For example, here are the average sample number and the *90th* percentile of the sample size for eight values of theta for design `tutorial.obf`.

theta	asn	X90
0.02	224.92	311.34
-0.05	284.10	389.17
-0.07	301.04	389.17
-0.09	307.87	389.17
-0.11	307.87	389.17
-0.13	301.04	389.17
-0.15	284.10	389.17
-0.22	224.91	311.34

From the command line, you can print the same information using the `seqOperatingChar` and `seqSampleSizeQuantile` functions.

Plotting Sample Size Distributions

You can plot the sample size distribution over a range of hypothesized values of the treatment effect using the **Average Sample Number** option on the **Plot** tab. The *75th* percentile is also plotted. For example, Figure 6.10 shows the ASN curves for the design `tutorial.obf`.

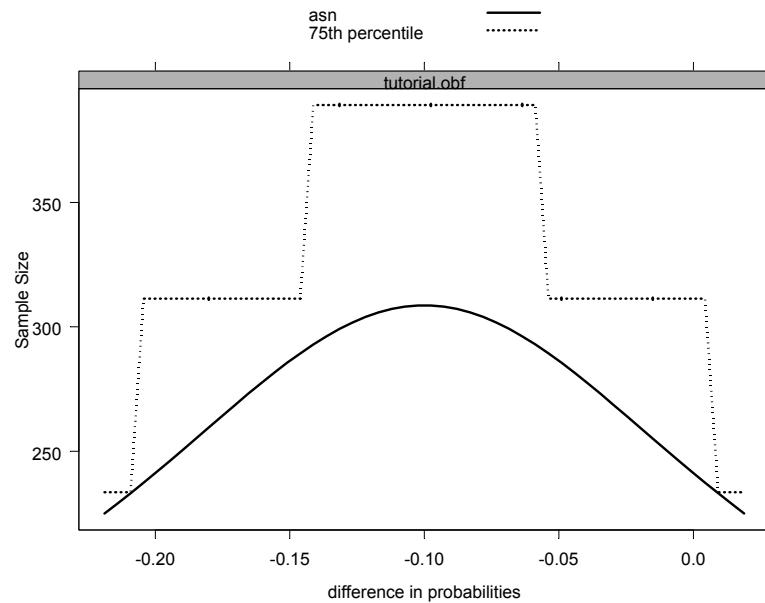


Figure 6.3: The ASN curves for the design `tutorial.obf`.

From the command line, you can plot sample size distributions using the `seqPlotASN` function. For example:

```
> seqPlotASN(tutorial.obf, fixed=F)
```

The **Overlay Designs** option on the **Plot** tab allows you to compare ASN curves for different designs by superposing the curves for the designs on the same plot. For example, Figure 6.4 overlays the ASN curves for designs `tutorial.obf` and `tutorial.poc` produced in Chapter 2.

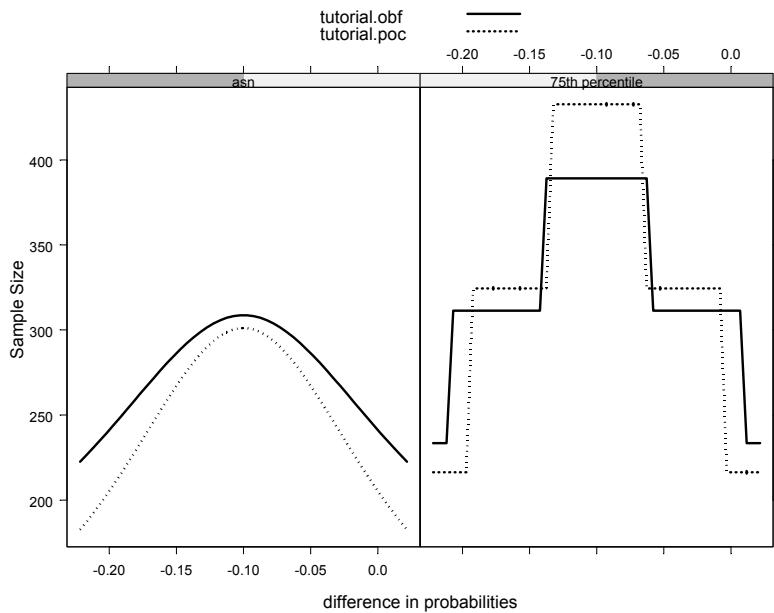


Figure 6.4: The ASN curves for designs `tutorial.obf` and `tutorial.poc`.

From the command line, you can overlay ASN curves using the `superpose.design=T` argument to the `seqPlotASN` function. For example:

```
> seqPlotASN(tutorial.obf, tutorial.poc,  
+           superpose.design=T, fixed=F)
```

POWER FUNCTIONS

In classical Neyman-Pearson hypothesis testing, critical values are chosen for rejecting the null hypothesis so that the probability of falsely rejecting the null (referred to as the Type I statistical error) is acceptably low. The accepted level of Type I error is called the *significance level*, or just the level of the test.

It is often more convenient, however, to consider the probability of rejecting the null hypothesis under various hypothesized treatment effects. The *power function* of the test $\beta(\theta_0)$ is the probability of rejecting the null hypothesis as a function of the true value of the unknown mean. The Type I error is then the value of the power function when the null hypothesis is true, $\alpha = \beta(\theta_0)$, and the Type II error (the probability of falsely failing to reject the null) for some given value of the unknown mean θ is $1 - \beta(\theta)$.

In S+SEQTRIAL, the stopping sets consistent with rejection of the null hypothesis vary in structure according to whether you are testing one-sided or two-sided hypotheses. In defining the operating characteristics of a group sequential test, it is therefore more useful to define two functions. An *upper* power function $\beta_+(\theta)$ measures the probability of rejecting the null hypothesis H_0 and the lower alternative hypothesis H_- in favor of the upper alternative H_+ . A *lower* power function $\beta_-(\theta)$ measures the probability of rejecting the null hypothesis H_- and the upper alternative hypothesis H_+ in favor of the lower alternative H_- . The upper power function is considered primarily in one-sided hypothesis tests of a greater alternative or in a two-sided test when choosing sample sizes based on a higher alternative. Similarly, the lower power function is considered primarily in one-sided hypothesis tests of a lesser alternative or in a two-sided test when choosing sample sizes based on a lower alternative.

You can use the power curve to determine the scientific interpretations that can correspond to a negative study. For example, if you are using a 95% confidence level as your criteria for statistical evidence, then by examining the alternative for which your design has 97.5% power, you can determine the scientific conclusion that

would correspond to a failure to reject the null hypothesis—a failure to reject the null hypothesis corresponds to a certain rejection of the alternative hypothesis for which your study has 97.5% power. In general, when using a confidence level of $1 - \alpha$ as the statistical criterion for evidence (that is, in a one-sided level $\alpha/2$ hypothesis test, or a two-sided level α hypothesis test), the interpretation of a negative study is certain to reject the alternative for which your study has power $1 - \alpha/2$.

Tabulating Power Functions

You can tabulate the lower and upper power functions for five default values of the treatment effect using the **Summary Tables** option on the **Results** tab. For example, here is part of the summary output for the design tutorial.obf produced in Chapter 2. Note that this design is one-sided test.

```
Operating characteristics
Theta      ASN Power.lower
-0.20  241.1993    0.9750
-0.15  286.3289    0.8371
-0.10  308.6648    0.5000
-0.05  286.3294    0.1629
  0.00  241.1998    0.0250
```

From the command line, you can print the same information using the `summary` function. For example:

```
> summary(tutorial.obf)
```

You can tabulate the lower and upper power functions for any number of hypothesized values of the treatment effect using the **Saved Results** groupbox on the **Results** tab. For example, here are eight values of the operating characteristics for design tutorial.obf.

```
Operating characteristics
Theta      ASN Power.lower
  0.0189  224.9158    0.01
-0.0472  284.0968    0.15
-0.0719  301.0436    0.29
-0.0910  307.8658    0.43
-0.1090  307.8657    0.57
-0.1281  301.0436    0.71
-0.1528  284.0968    0.85
-0.2189  224.9134    0.99
```

From the command line, you can print the same information using the seqOperatingChar function. For example:

```
> seqOperatingChar(tutorial.obf, how.many=8)
```

Plotting Power Functions

You can plot the power function over a range of hypothesized values of the treatment effect (theta) using the **Power Curve** option on the **Plot** tab. For example, Figure 6.5 shows the power curve for design tutorial.obf.

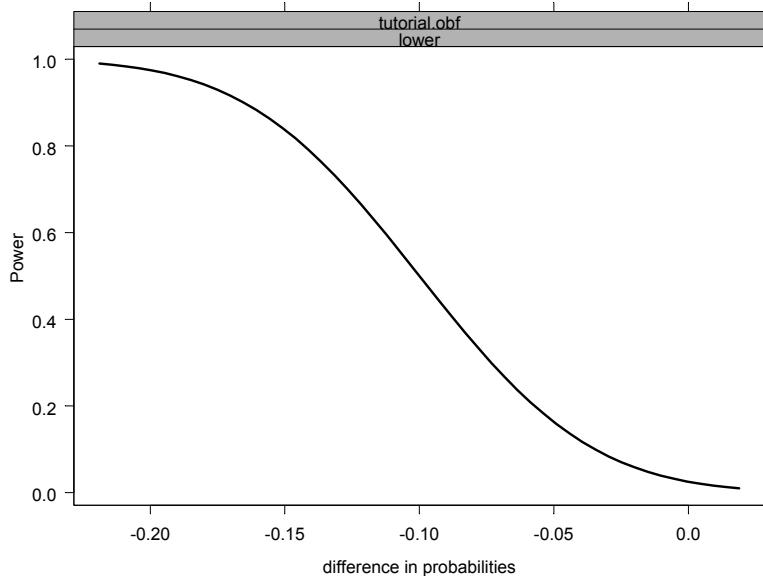


Figure 6.5: The power curve for the design tutorial.obf.

From the command line, you can plot power curves using the seqPlotPower function. For example:

```
> seqPlotPower(tutorial.obf)
```

The **Overlay Designs** option on the **Plot** tab allows you to compare power curves for several different designs by superposing the power curves for the designs on the same plot. Choose one or more designs to compare to the current design from the **Designs to Plot** listbox, select the **Overlay Designs** option, then click **Apply** or **OK**. For example, Figure 6.6 overlays the power curves for designs `tutorial.obf` and `tutorial.poc` produced in Chapter 2.

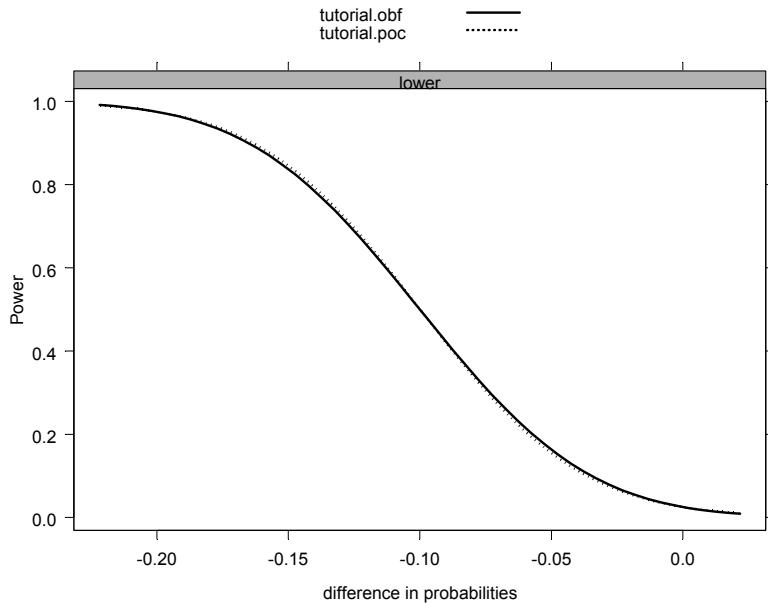


Figure 6.6: The power curves for designs `tutorial.obf` and `tutorial.poc`.

From the command line, you can overlay power curves using the `superpose.design=T` argument to the `seqPlotPower` function. For example:

```
> seqPlotPower(tutorial.obf, tutorial.poc,
+               superpose.design=T)
```

Direct comparisons are sometimes clearest if you consider the difference in power between candidate designs and some reference design. This can be achieved by selecting the **Power Curve vs.**

Reference plot option on the **Plot** tab. Specify the reference design (the design subtracted from all others) using the **Reference Design** field in the **Options** groupbox. For example, Figure 6.7 plots the power curve for design `tutorial.obf` against reference design `tutorial.poc`.

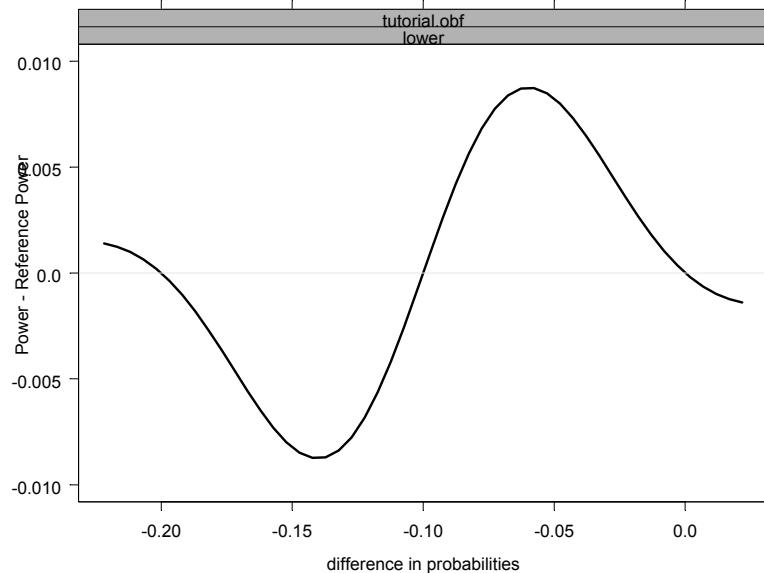


Figure 6.7: Plot of the power curve for design `tutorial.obf` against reference design `tutorial.poc`.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(tutorial.obf, reference=tutorial.poc)
```

STOPPING PROBABILITIES

The power function described above applies equally well to both the fixed sample and group sequential settings. In the group sequential setting, however, it is often of interest to consider the probability of stopping at each analysis. S+SEQTRIAL allows you to examine the probability of making each decision at each analysis time as the treatment effect varies from harm to equivalence to benefit.

Tabulating Stopping Probabilities

You can tabulate the stopping probabilities for five default values of the treatment effect (theta) using the **Summary Tables** option on the **Results** tab. For example, these are the stopping probabilities for the design tutorial.obf produced in Chapter 2.

Stopping Probabilities:

Theta	Time 1	Time 2	Time 3	Time 4	Time 5
-0.20	0.0035	0.2595	0.4431	0.2226	0.0714
-0.15	0.0008	0.1061	0.3296	0.3404	0.2230
-0.10	0.0003	0.0558	0.2476	0.3704	0.3259
-0.05	0.0008	0.1061	0.3296	0.3404	0.2230
0.00	0.0035	0.2595	0.4431	0.2226	0.0714

From the command line, you can print the same information using the summary function. For example:

```
> summary(tutorial.obf)
```

You can tabulate the probability of stopping for the lower alternative hypothesis, null hypothesis, and upper alternative hypothesis at each analysis for any number of hypothesized values of the treatment effect using the **Saved Results** groupbox on the **Results** tab.

Plotting Stopping Probabilities

You can plot the stopping probabilities over a range of hypothesized values for the treatment effect using the **Stopping Probability** option on the **Plot** tab. For example, Figure 6.8 shows the stopping probabilities for design tutorial.obf.

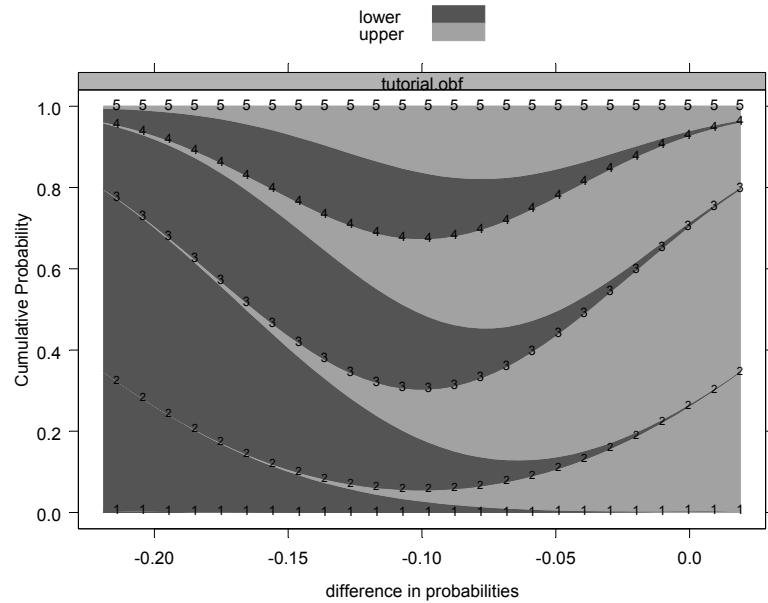


Figure 6.8: Stopping probabilities for the design `tutorial.obf`.

From the command line, you can plot stopping probabilities using the `seqPlotStopProb` function. For example:

```
> seqPlotStopProb(tutorial.obf)
```

THE RESULTS TAB

The **Results** tab, shown in Figure 6.9, controls the printed output from S+SEQTRIAL.

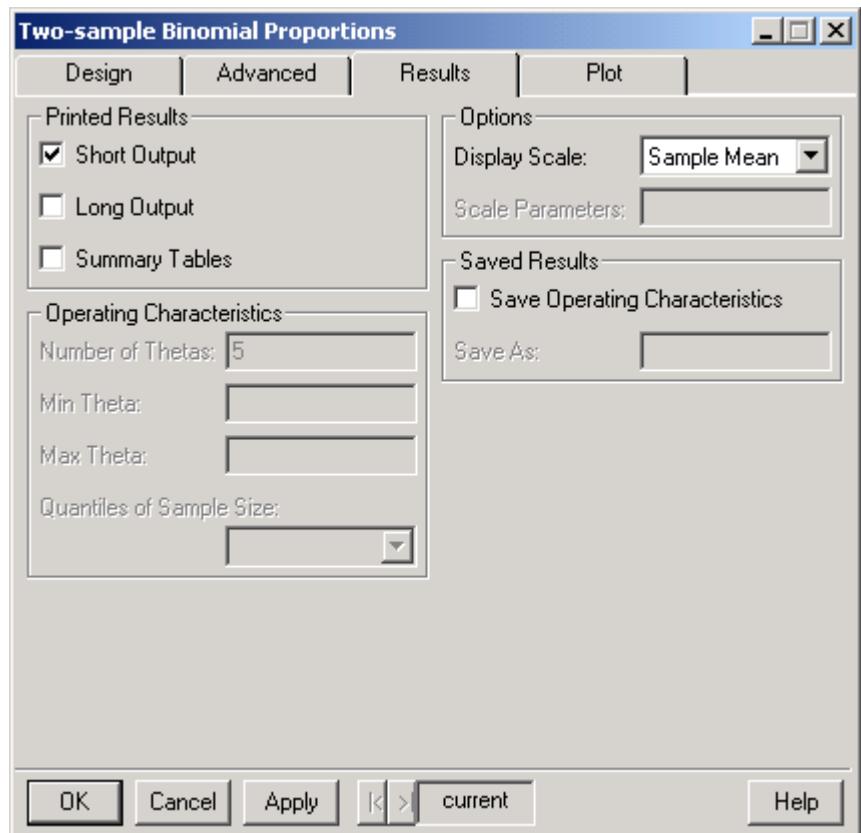


Figure 6.9: The **Results** tab of the S+SEQTRIAL dialog.

The **Operating Characteristics** groupbox is inactive until you select either **Summary Tables** or **Save Operating Characteristics**.

Short Output

The **Short Output** option is the default output from S+SEQTRIAL. It begins by reporting the call to the underlying `seqDesign` function that was made when you clicked **OK** or **Apply**. For example, this is the call for the design `tutorial.ofb` created in Chapter 2.

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, log.transform =  
  F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,  
  0.6), variance = "alternative", nbr.analyses = 5,  
  test.type = "less", size = 0.025, power = 0.975)
```

It is often useful to examine this call to confirm that all parameters are being passed to `seqDesign` as you intend. You can also cut-and-paste this text into the **Commands** window to repeat the call.

Next, the probability model for your design and the operating characteristics of the hypothesis tests are reported.

```
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
One-sided hypothesis test of a lesser alternative:  
  
Null hypothesis : Theta >= 0      (size = 0.025 )  
Alternative hypothesis : Theta <= -0.2    (power = 0.975 )  
  
(Emerson & Fleming (1989) symmetric test)
```

Finally, the boundary values are reported on the current display scale, for each of the analyses. (See page 136 for instructions on how to change the display scale.) The sample size is also reported for each analysis. In a fixed sample design, where only a single analysis is specified, this is identified as Time 1 in the printed output. In a group sequential design, the multiple analyses are identified as Time 1 through Time k. For example:

```
STOPPING BOUNDARIES: Sample Mean scale  
          a      d  
Time 1 (N= 77.83) -0.5000  0.3000  
Time 2 (N= 155.67) -0.2500  0.0500  
Time 3 (N= 233.50) -0.1667 -0.0333  
Time 4 (N= 311.34) -0.1250 -0.0750  
Time 5 (N= 389.17) -0.1000 -0.1000
```

This example is a two-boundary design that uses the sample mean display scale. This design has 5 analyses, and can stop early after accrual of 78, 156, 234, 312 subjects (39, 78, 117, or 156 per treatment arm). The maximal sample size, achieved when the trial continues to the final analysis, is 390.

From the command line, you can print the same information using the `print` function:

```
> print(tutorial.obf)
```

Or just type the object name:

```
> tutorial.obf
```

Long Output

You can use the **Long Output** option to print a more verbose summary of a design. For example, this is the **Long Output** for the design `tutorial.obf`. Note the more detailed characterization of the stopping boundaries and interim sample sizes.

```
Call:
seqDesign(prob.model = "proportions", arms = 2, log.transform =
F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,
0.6), variance = "alternative", nbr.analyses = 5,
test.type = "less", size = 0.025, power = 0.975)

PROBABILITY MODEL and HYPOTHESES:

PROBABILITY MODEL:
Two arm study of binary response variable

Randomization scheme: 1 treatment group : 1 comparison group

Outcome summarized by event probability
Treatment Comparison
Null          0.6          0.6
Alternative    0.4          0.6

Treatment effect is difference in probabilities (Treatment -
Comparison)
Theta
Null          0.0
Alternative   -0.2

(Standardization parameters: Mu0 0; SigmaSqr 0.48; Psi 1)
```

Chapter 6 Evaluating Designs

HYPOTHESES:

Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:

Null hypothesis : Theta >= 0 (size = 0.025)
Alternative hypothesis : Theta <= -0.2 (power = 0.975)

Boundary hypotheses:

Boundary a rejects : Theta >= 0.0 (lower size = 0.025)
Boundary b rejects : Theta <= -0.2 (lower power = 0.975)
Boundary c rejects : Theta >= 0.0 (upper power = 0.975)
Boundary d rejects : Theta <= -0.2 (upper size = 0.025)

FULL PARAMETERIZATION: (Emerson & Fleming (1989) symmetric test)
Number and timing of analyses:

	Time 1	Time 2	Time 3	Time 4	Time 5
Combined sample size	77.83	155.67	233.5	311.3	389.2
(Treatment arm)	38.92	77.83	116.8	155.7	194.6
(Comparison arm)	38.92	77.83	116.8	155.7	194.6
Cum proportion	0.20	0.40	0.6	0.8	1.0

Size, power, and hypothesis shift parameters:

	Lower	Upper
Alpha	0.025	0.025
Beta	0.975	0.975
Epsilon	1	0

Alpha (size = 0.025)
Beta (type I and type II errors equal)
Epsilon (one-sided test of lesser alternative)

Design family based on: Sample Mean scale

Boundary shape parameters:

P	A	R	G		
a	1	0	0	2.0134	(O'Brien-Fleming)
b	1	0	0	2.0134	(O'Brien-Fleming)
c	1	0	0	2.0134	(O'Brien-Fleming)
d	1	0	0	2.0134	(O'Brien-Fleming)

Constraints -

Exact constraints: (none)

Minimum constraints: (none)

Maximum constraints: (none)

STOPPING BOUNDARIES: Sample Mean scale

a	d
Time 1 (N= 77.83)	-0.5000 0.3000

```
Time 2 (N= 155.67) -0.2500 0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000
```

STOPPING BOUNDARIES: Error Spending Function scale

	a	d
Time 1 (N= 77.83)	0.0001	0.0001
Time 2 (N= 155.67)	0.0292	0.0292
Time 3 (N= 233.50)	0.1974	0.1974
Time 4 (N= 311.34)	0.5542	0.5542
Time 5 (N= 389.17)	1.0000	1.0000

STOPPING BOUNDARIES: Standardized Cumulative Sum scale

	a	d
Time 1 (N= 0.2)	-2.0134	1.2081
Time 2 (N= 0.4)	-2.0134	0.4027
Time 3 (N= 0.6)	-2.0134	-0.4027
Time 4 (N= 0.8)	-2.0134	-1.2081
Time 5 (N= 1.0)	-2.0134	-2.0134

From the command line, you can print the same information using the `details=T` argument to the `print` function:

```
> print(tutorial.ofb, details=T)
```

Summary Tables

You can use the **Summary Tables** option to print advanced output from S+SEQTRIAL. When you click on the **Summary Tables** checkbox, the fields in the **Operating Characteristics** groupbox are activated. You can use these fields to specify the values of theta (the treatment effect) at which the operating characteristics are computed. These values are determined by the **Number of Thetas**, the **Min Theta**, and the **Max Theta** fields shown in Figure 6.9. Operating characteristics are computed for the specified number of theta values equally spaced between the specified minimum and maximum. If **Min Theta** and **Max Theta** are left unspecified, the values of theta corresponding to the null and alternative hypotheses are used as the limits.

The operating characteristics computed include

- The lower and upper power for each value of theta.

- The average sample number for each value of theta. Use the **Quantiles of Sample Size** field to save additional vectors containing the desired quantiles of the sample size.
- The stopping probabilities for the lower alternative hypothesis, null hypothesis, and upper alternative hypothesis for each value of theta.

For example, selecting the **Summary Tables** checkbox with default values for **Number of Thetas** when creating the tutorial.ofb design would result in the following output in the **Report** window:

```
Operating characteristics
Theta      ASN Power.lower
-0.20    241.1993    0.9750
-0.15    286.3289    0.8371
-0.10    308.6648    0.5000
-0.05    286.3294    0.1629
  0.00    241.1998    0.0250
```

Next, the probability of stopping at each analysis are tabulated for five hypothesized values of the treatment effect (theta).

```
Stopping Probabilities:
Theta Time 1 Time 2 Time 3 Time 4 Time 5
-0.20 0.0035 0.2595 0.4431 0.2226 0.0714
-0.15 0.0008 0.1061 0.3296 0.3404 0.2230
-0.10 0.0003 0.0558 0.2476 0.3704 0.3259
-0.05 0.0008 0.1061 0.3296 0.3404 0.2230
  0.00 0.0035 0.2595 0.4431 0.2226 0.0714
```

The operating characteristics are also saved as a data frame.

Point estimates, *p*-values, and confidence intervals are then estimated for each boundary at each analysis. Four different point estimates are given:

- MLE, maximum likelihood estimate based on the observed statistics;
- BAM, bias-adjusted mean estimate;
- RBadj, Rao-Blackwell adjusted estimate (under some conditions this is a UMVUE, uniformly minimum-variance unbiased estimate);
- MUE, median-unbiased estimate.

For example:

		Inferences at the Boundaries		
		*** a Boundary *** *** d Boundary ***		
Time 1		Boundary	-0.500	0.300
	MLE	-0.500	0.300	
	BAM	-0.478	0.278	
	RBadj	-0.500	0.300	
	MUE	-0.472	0.272	
	P-value	0.000	0.997	
95% Conf Int		(-0.639, -0.280)	(0.080, 0.439)	
Time 2		Boundary	-0.250	0.050
	MLE	-0.250	0.050	
	BAM	-0.234	0.034	
	RBadj	-0.250	0.050	
	MUE	-0.233	0.033	
	P-value	0.001	0.682	
95% Conf Int		(-0.365, -0.093)	(-0.107, 0.165)	
Time 3		Boundary	-0.167	-0.033
	MLE	-0.167	-0.033	
	BAM	-0.155	-0.045	
	RBadj	-0.163	-0.037	
	MUE	-0.154	-0.046	
	P-value	0.005	0.212	
95% Conf Int		(-0.266, -0.037)	(-0.163, 0.066)	
Time 4		Boundary	-0.125	-0.075
	MLE	-0.125	-0.075	
	BAM	-0.120	-0.080	
	RBadj	-0.118	-0.082	
	MUE	-0.115	-0.085	
	P-value	0.015	0.049	
95% Conf Int		(-0.216, -0.011)	(-0.189, 0.016)	
Time 5		Boundary	-0.100	-0.100
	MLE	-0.100	-0.100	
	BAM	-0.100	-0.100	
	RBadj	-0.100	-0.100	
	MUE	-0.100	-0.100	
	P-value	0.025	0.025	
95% Conf Int		(-0.200, 0.000)	(-0.200, 0.000)	

This example shows a two-boundary design with five analyses. See Chapter 8, Reporting Results, for more information on reporting inferences adjusted for the presence of a stopping rule.

From the command line, you can print the same information using the `summary` function. For example:

```
> summary(tutorial.obf)
```

Options

As discussed in Chapter 5 (see page 112), different authors define their group sequential boundaries on different boundary scales. The **Display Scale** field determines how S+SEQTRIAL functions report group sequential boundaries. Select from among the supported boundary scales discussed on page 112.

From the command line, specify the display scale using the `display.scale` argument to the `seqDesign` function.

The **Scale Parameters** field allows you to enter additional parameters for the Error Spending, Bayesian, Conditional Futility, and Predictive Futility scales, and is greyed out unless one of these scales is selected in the **Display Scale** field. See page 140 for more information.

From the command line, specify the scale parameters as arguments to the `seqScale` function.

Saved Results

The **Saved Results** groupbox allows the operating characteristics to be saved in a named data frame.

THE PLOT TAB

The **Plot** tab, shown in Figure 6.10, controls plotting from S+SEQTRIAL.

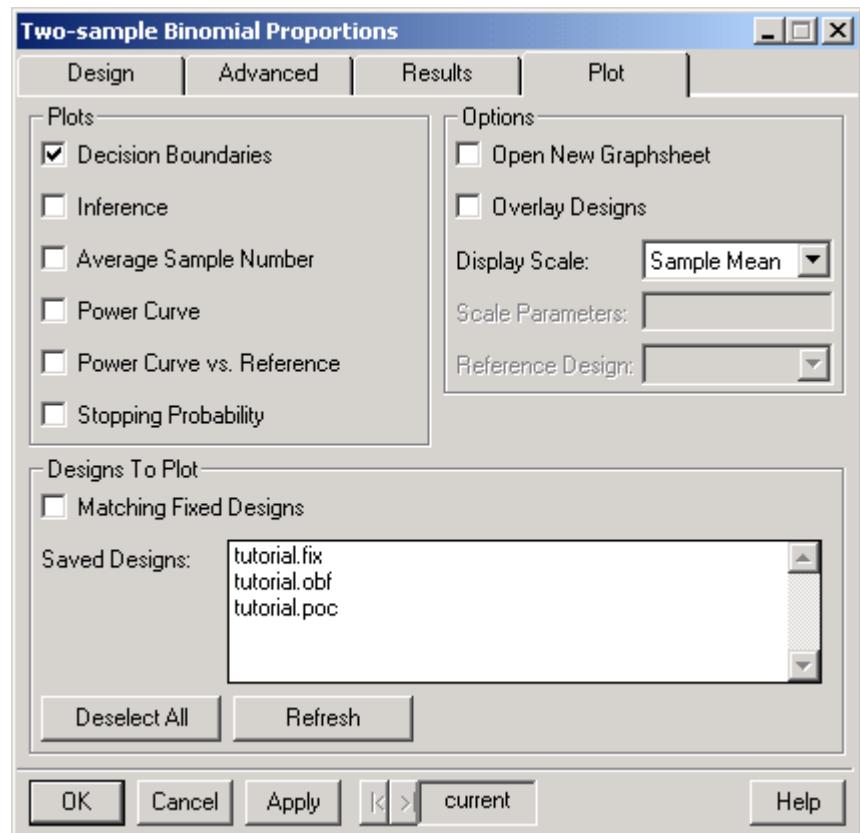


Figure 6.10: Sample **Plot** tab from the S+SEQTRIAL dialog.

Plot Types

S+SEQTRIAL offers a variety of difference plot types for evaluating your designs.

Decision Boundaries

The **Decision Boundaries** option plots the stopping boundaries for each analysis on the current **Display Scale**. See page 164 for a discussion of evaluating stopping boundaries in printed and plotted output.

From the command line, you can plot stopping boundaries using the `seqPlotBoundary` function, or by calling `plot` (which calls `plot.seqDesign`, which calls `seqPlotBoundary`).

Inference

The **Inference** option plots the point estimates and 95% confidence intervals for each boundary at each analysis. See Chapter 8, Reporting Results, for a discussion of evaluating inferences on the boundaries in printed and plotted output.

From the command line, you can plot inferences at the boundaries using the `seqPlotInference` function.

Average Sample Number (ASN)

The **Average Sample Number** option plots the sample size distribution over a range of hypothesized values of the treatment effect. The 75th percentile is also plotted. See page 176 for a discussion of evaluating sample size distributions in printed and plotted output.

From the command line, you can plot ASN curves using the `seqPlotASN` function.

Power Curve

The **Power Curve** option plots the lower and upper power functions over a range of hypothesized values of the treatment effect. See page 180 for a discussion of evaluating power functions in printed and plotted output.

From the command line, you can plot power curves using the seqPlotPower function.

Power Curve vs. Reference

The **Power Curve vs. Reference** option plots the difference between power curves. This option is useful for comparing designs because it is often the case that the power curves are visually almost indistinguishable. Specify the reference design (the design subtracted from all others) using the **Reference Design** field in the **Options** groupbox.

From the command line, plot the difference between power curves using the reference argument to the seqPlotPower function.

Stopping Probabilities

The **Stopping Probabilities** option plots the cumulative stopping probabilities for the lower alternative, null, and upper alternative hypotheses over a range of hypothesized values of the treatment effect. See page 185 for a discussion of evaluating stopping probabilities in printed and plotted output.

From the command line, you can plot stopping probabilities using the seqPlotStopProb function.

Designs to Plot The **Designs to Plot** groupbox allows you to select other designs to compare to the current design. If the **Overlay Designs** option is selected, the comparison design plots are superimposed on the same panel with the current design; otherwise, they are plotted on adjacent panels.

This list of designs in the **Designs to Plot** groupbox is constructed when the S+SEQTRIAL dialog is first launched. To update the list, press the **Refresh** button. To deselect all items in the list, press the **Deselect All** button.

From the command line, all S+SEQTRIAL plotting functions can take multiple design arguments. For example

```
> seqPlotASN(tutorial.obf, tutorial.poc)
```

Use the `superpose.design=T` argument to overlay the plots.

The **Matching Fixed Designs** option causes S+SEQTRIAL to plot the corresponding fixed sample designs matched for power or maximal sample size, depending on the plot. For example, using the **Power Curve** plot type, you could compare the gain and/or loss in power under various group sequential stopping rules relative to the power of a fixed sample design which uses the same maximal number of subjects. In this way, you could estimate the cost of adding interim analyses to a fixed sample study. Using the **Decision Boundaries** and **Average Sample Number** plot type, you could compare the stopping boundaries and average efficiency of group sequential stopping rules to the corresponding values for a fixed sample test which has the same power. The **Matching Fixed Designs** option only applies to the **Decision Boundaries**, **Average Sample Number**, and **Power Curve** plot types, and is greyed out unless one of these plot types is selected.

From the command line, plot the matched fixed sample designs using the `fixed=T` argument to the plotting function:

- For function `seqPlotBoundary`, the `fixed` argument defaults to `TRUE` when a single design is plotted, and to `FALSE` when more than one design are plotted;
- For the `seqPlotASN` function, the `fixed` argument defaults to `TRUE`;
- For the `seqPlotPower` function, the `fixed` argument defaults to `FALSE`. An additional argument governs whether the fixed design is chosen to match the power (at the alternative hypothesis) or maximal sample sizes; the default is `match.power=FALSE` (for the boundary and ASN plotting functions, the fixed design always matches the power).

Options

The **Options** groupbox contains plotting options that can be applied to particular plot types.

Open New Graphsheet

The **Open New Graphsheet** option causes S+SEQTRIAL to open a new graphsheet before plotting. This prevents S+SEQTRIAL from overwriting an existing plot, but can also quickly multiply the number of open graphsheets in your session, since a new graphsheet is created every time you click **OK** or **Apply**.

From the command line, open a new graphsheet using the `graphsheet` command. Subsequent plots are drawn on the new panel.

Overlay Designs

The **Overlay Designs** option causes S+SEQTRIAL to overlay plots on the same panel. The current design is always plotted; select additional comparison designs from the **Designs to Plot** groupbox (see above).

From the command line, overlay designs using the `superpose.design=T` argument to any of the S+SEQTRIAL plotting commands.

Display Scale

As discussed in Chapter 5 (see page 112), different authors define their group sequential boundaries on different boundary scales. The **Display Scale** field determines how S+SEQTRIAL functions plot group sequential boundaries. Select from among the supported boundary scales discussed on page 112. This option is only available for the **Decision Boundaries** plot type.

From the command line, specify the display scale using the `display.scale` argument to the `seqDesign` function.

Scale Parameters

The **Scale Parameters** field allows you to enter additional parameters for the Error Spending, Bayesian, Conditional Futility, and Predictive Futility scales, and is greyed out unless one of these scales is selected in the **Display Scale** field. See page 140 for more information.

From the command line, specify the scale parameters as arguments to the `seqScale` function.

Reference Design

The **Reference Design** field allows you to specify the design whose power curve is subtracted from all others in the **Power Curve vs. Reference** plot type.

From the command line, plot the difference between power curves using the `reference` argument to the `seqPlotPower` function.

MONITORING A TRIAL

7

Overview	202
Reasons for Monitoring a Clinical Trial	203
Statistical Issues in Monitoring a Clinical Trial	204
Computing Modified Boundaries	206
Constrained Boundaries on the Sample Mean Scale	208
Constrained Boundaries on the Error Spending Scale	215
Additional Examples	222
Specifying a Monitoring Procedure	223
Previous Design	223
Data	224
Future Analyses	224
Constraint Scale	225
Minimum Spacing of Analyses	226
Maximum Iterations	226
Evaluating A Monitoring Rule	227

OVERVIEW

When you are satisfied with your design, you can begin to collect data. In a group sequential design, you must monitor the trial to determine if early stopping is appropriate. S+SEQTRIAL includes a monitoring process for this purpose.

The monitoring process also computes modified stopping boundaries based on revised estimates of response measurement variability, taking into account whether the number and timing of analyses that occur during the actual conduct of a group sequential trial vary from the number and timing of analyses estimated during the design phase.

This chapter describes:

- The reasons for monitoring a group sequential trial;
- Statistical issues in monitoring a clinical trial;
- How to monitor a group sequential trial.

The basic command line function for clinical trial design in S+SEQTRIAL is `seqMonitor`. The `update` function can be used to modify an existing monitoring object.

REASONS FOR MONITORING A CLINICAL TRIAL

There are three main reasons to monitor a clinical trial while it is in progress:

1. To ensure that the patients enrolled in the clinical trial are treated ethically;
2. To ensure that future patients not enrolled in the clinical trial receive the best possible treatment;
3. To increase the efficiency of the process whereby new treatments are adopted.

The first two ethical issues are of paramount interest to the Data Monitoring Committee. This third issue is indirectly ethical in nature: More efficient processes make more effective treatments available to more people sooner.

STATISTICAL ISSUES IN MONITORING A CLINICAL TRIAL

The stopping rule chosen in the design phase serves as a guideline to a Data Monitoring Committee as it makes a decision recommending continuing or stopping a clinical trial. If all aspects of the conduct of the clinical trial adhered exactly to the conditions stipulated during the design phase, the stopping rule obtained during the design phase could be used directly. However, there are usually at least two complicating factors that must be dealt with during the conduct of the clinical trial:

- The schedule of interim analyses may not follow that used in the design of the trial. Meetings of the Data Monitoring Committee are typically scheduled according to calendar time, so the statistical information (sample sizes) available for analysis at any given meeting is a random variable. Similarly, accrual may be slower or faster than planned, resulting in a different number of interim analyses than originally planned. Either of these eventualities require modification of the stopping rule, because the exact stopping boundaries are dependent upon the number and timing of analyses. For example, an O'Brien-Fleming (1979) design for four equally spaced analyses has different stopping thresholds than an O'Brien-Fleming (1979) design for four analyses scheduled after 26%, 51%, 76%, and 100% of the data are accrued.
- The estimate of response variability used during the design phase may be incorrect. Crude estimates of response variability or baseline event rates are often used during the design phase. As the trial progresses, more accurate estimates are available. Clearly the operating characteristics of particular stopping rules are heavily dependent on the variability of response measurement.

In order to address these issues, flexible methods of implementing stopping rules have been developed that allow the clinical trialist to maintain at least some of the operating characteristics of the stopping rule. Typically, such flexible methods always maintain the size (Type I error) at the prescribed level. A choice must then be made as to whether the maximal sample size or the power to detect the design alternative should be maintained.

The flexible methods of implementing stopping rules provided by S+SEQTRIAL are based on the idea of computing a stopping boundary for the current interim analysis so that the desired operating characteristics are satisfied, and the stopping rule is constrained to agree with the stopping boundaries used at all previously conducted interim analyses. Thus, the flexible monitoring methods are based on the concept of the constrained stopping boundaries described in Chapter 5 on page 158. These methods include the error spending function approach as a special case.

COMPUTING MODIFIED BOUNDARIES

Prior to monitoring a clinical trial, you must have stored:

- a **seqDesign** object which represents the stopping rule to be used in the monitoring process. (**Chapter 4, Specifying a Fixed Sample Design** and **Chapter 5, Specifying a Group Sequential Design** describe the process of creating such a **seqDesign** object.)
- a dataframe containing a vector of response measurements and a vector of treatment assignments for patients accrued to the study. The response measurements should be appropriate for the probability model specified in the **seqDesign** object, and the treatment variable should be coded such that the comparison group is 0 (or FALSE), and in a two arm study the new treatment group is 1 (or TRUE).

For example, in this section we consider the setting of monitoring a clinical trial in the hypothetical setting used in **Chapter 2, Tutorial**. In that tutorial, we consider a randomized, double blind, placebo controlled two arm clinical trial in which the primary endpoint is 28 day mortality, a binary endpoint. The measure of treatment effect is the difference in 28 day mortality rates. Patients treated with placebo are expected to experience 60% mortality, while it is hoped that patients receiving the new treatment experience only 40% mortality during the first 28 days post randomization.

Suppose that the design **tutorial.obf** is chosen for the conduct of this clinical trial. Thus, during the planning of the trial we envision a one-sided level 0.025 hypothesis test of the null hypothesis of no difference between the treatment arms. We are interested in the lesser alternative hypothesis that the difference in 28 day mortality would be -0.2 (new treatment mortality minus placebo mortality), and we desire 97.5% power to detect this alternative (so we choose the type II error equal to the type I error in order that a 95% confidence interval would discriminate between the null and alternative hypotheses). We plan on performing 5 equally spaced analyses using a one-sided symmetric group sequential stopping rule with O'Brien-Fleming boundary relationships. As shown in the S+SEQTRIAL output below, when response variability is computed under the alternative hypothesis and patients are randomized equally to the two treatment

arms, such a design estimates that a maximal sample size of 389 subjects (approximately 195 subjects per arm) need to be accrued to the study in order to provide the desired precision. Under this plan, interim analyses are performed after the accrual of approximately 78, 156, 234, and 311 patients to the study.

```
Call:
seqDesign(prob.model = "proportions", arms = 2, log.transform =
F, null.hypothesis = c(0.6, 0.6), alt.hypothesis = c(0.4,
0.6), variance = "alternative", nbr.analyses = 5,
test.type = "less", size = 0.025, power = 0.975)

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
Null hypothesis : Theta >= 0      (size = 0.025)
Alternative hypothesis : Theta <= -0.2    (power = 0.975)
(Emerson & Fleming (1989) symmetric test)

STOPPING BOUNDARIES: Sample Mean scale
                     a      d
Time 1 (N= 77.83) -0.5000  0.3000
Time 2 (N= 155.67) -0.2500  0.0500
Time 3 (N= 233.50) -0.1667 -0.0333
Time 4 (N= 311.34) -0.1250 -0.0750
Time 5 (N= 389.17) -0.1000 -0.1000
```

In practice, the data frame containing the data often is created by importing data into S-PLUS from another source, for example an Excel spreadsheet. Here we create artificial data to mimic the process of monitoring a clinical trial, starting with the first analysis. We suppose that the actual analyses occur with sample sizes somewhat different than originally planned, say 120 subjects (60 per arm). We create a variable **treatment1** using the following commands entered into a **Commands** window (**treatment1** should be 0 for the placebo group and 1 for the new treatment).

```
> treatment1 <- rep(0:1, each=60)
```

We then simulate the responses obtained for those 120 patients contributing information at the first interim analysis. Suppose that the true placebo 28 day mortality rate is 50%, somewhat less than originally estimated when planning the trial, and suppose the new treatment is truly efficacious, having a 28 day mortality rate of 35%. The following code can be entered into a **Commands** window to

create a variable **response1** which contains the simulated data from the first 120 observations. (We set the seed for the random number generator in order to allow you to reproduce the results.)

```
> set.seed(0)
> response1 <- rbinom(120, 1, ifelse(treatment1 == 1, 0.35, 0.50))
```

We then create an S+ dataset named **firstData** containing these two variables.

```
> firstData <- data.frame(response1, treatment1)
```

Constrained Boundaries on the Sample Mean Scale

The S+SEQTRIAL monitoring process is then initiated by selecting **Sequential Monitor** from the first level of the **SeqTrial** menu hierarchy, as shown in Figure .

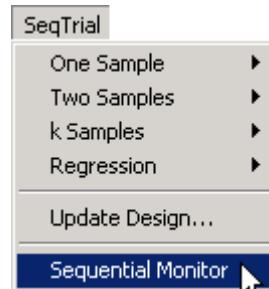


Figure 7.1: Using the SeqTrial menu hierarchy to monitor a clinical trial.

This launches the **Group Sequential Monitoring** dialog as shown in Figure 7.2.

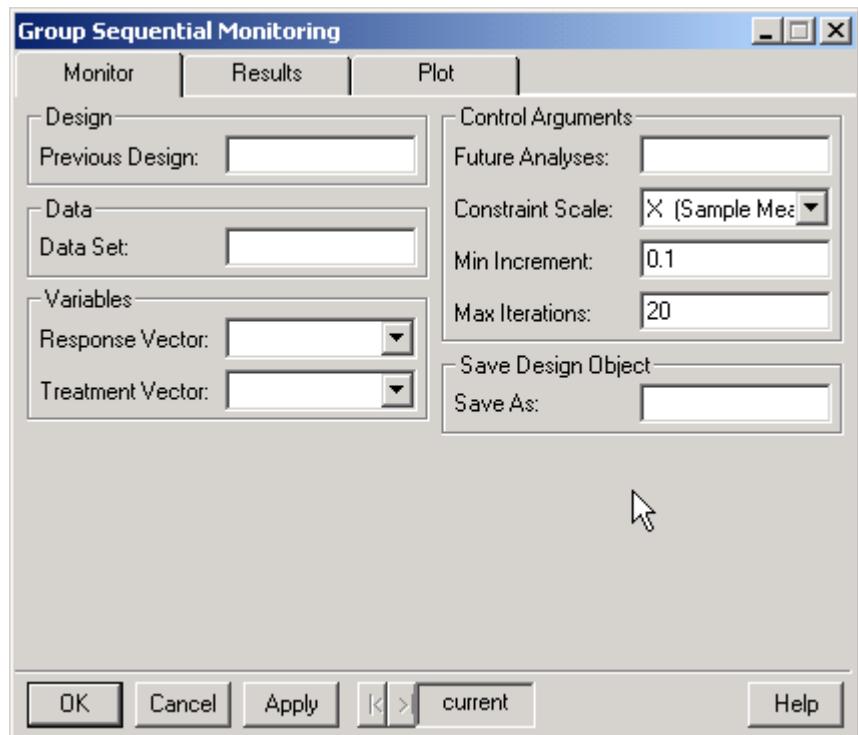


Figure 7.2: The *S+SEQTRIAL* dialog for group sequential monitoring of a clinical trial.

In order to perform the first interim analysis, we

- Fill in the **Previous Design** text box with the name of the **seqDesign** object containing the clinical trial design: **tutorial.obf**,
- Fill in the **Data Set** text box with the name of the data set containing the clinical trial data: **firstData**,
- Select the name of the vector recording the indicators of 28 day mortality for each patient: **response1**,
- Select the name of the vector indicating the treatment: **treatment1**,
- Use the default values for all the fields in the **Control Arguments** groupbox (these options are explained in the next section of this chapter), and

- Fill in a name to be used to store the **seqMonitor** object which results from this interim analysis: **monitor1**.

The completed **Monitor** tab of the **Group Sequential Monitoring** dialog now appears as shown in Figure 7.3.

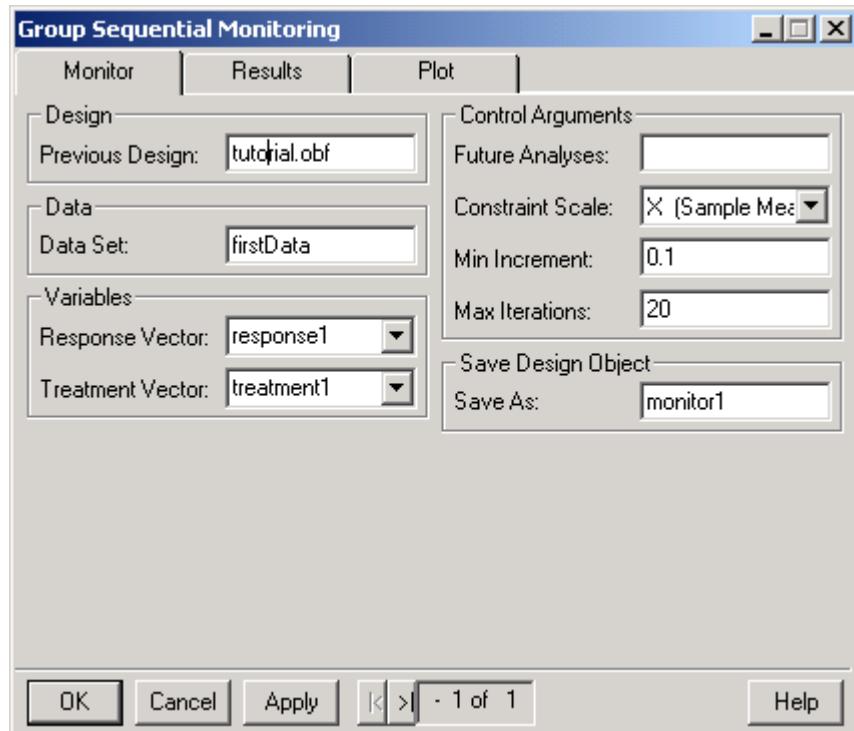


Figure 7.3: The completed dialog for the first interim analysis of the simulated data using **tutorial.obf**.

Upon clicking **OK**, S+SEQTRIAL uses the information stored in **tutorial.obf** about the stopping rule parameters (e.g., the test type, the level of significance, desired power under the alternative, the boundary shape function, and the planned monitoring schedule) along with the information in the dataset **firstData** (e.g., the sample size, the event rate in the two arms) to create an updated stopping rule. By default, the new monitoring schedule is based on the actual sample size available in the current data, with the timing of future analyses taken to be at the same proportions of maximal sample size as in the original schedule of analyses. Also by default, the maximal

sample size is re-estimated to provide the desired power to detect the alternative hypothesis. This re-estimation adjusts for the revised schedule of analyses as well as using the current estimate of the variability of the test statistic. The following output appears in the **Report** window.

```
*** Sequential Monitor ***
Call:
seqMonitor(x = tutorial.obf, response = firstData$response,
treatment = firstData$treatment1, constraint.scale = "X",
min.increment = 0.1, maxiter = 20)

RECOMMENDATION:
Continue

OBSERVED STATISTICS:
Sample Size Crude Estimate Z Statistic
120           -0.15      -1.664

MONITORING BOUNDS:
Call:
"(not shown, is (your seqMonitor object)$seqDesignCall)"

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
Null hypothesis : Theta >= 0      (size = 0.025)
Alternative hypothesis : Theta <= -0.1999   (power = 0.975)
(Emerson & Fleming (1989) symmetric test)

STOPPING BOUNDARIES: Sample Mean scale
                     a      d
Time 1 (N= 120) -0.3290  0.1291
Time 2 (N= 237) -0.1666 -0.0333
Time 3 (N= 316) -0.1249 -0.0750
Time 4 (N= 395) -0.0999 -0.0999
```

In this case, the first analysis occurring after accrual of 120 subjects was judged to replace both the first and second of the originally planned analyses. The increase in the maximal sample size (395 rather than 389) is largely due to an estimated event rate that was closer to 0.5 (and hence estimated with less precision) than was originally estimated.

From the above output, you can see that the observed difference in 28 day mortality is -0.15. According to the modified stopping rule, early termination of the study with a decision for efficacy would only occur

if that crude estimate of treatment effect were less than -0.329, and early termination with a decision for futility would only occur if the crude estimate of treatment effect were greater than 0.1291. Hence, the output provides the recommendation to continue.

A graphical display of the interim monitoring results can be obtained by selecting **Decision Boundaries** on the **Plots** tab of the **Group Sequential Monitoring** dialog, with **Overlay Designs** selected. Such a selection produces Figure 7.4.

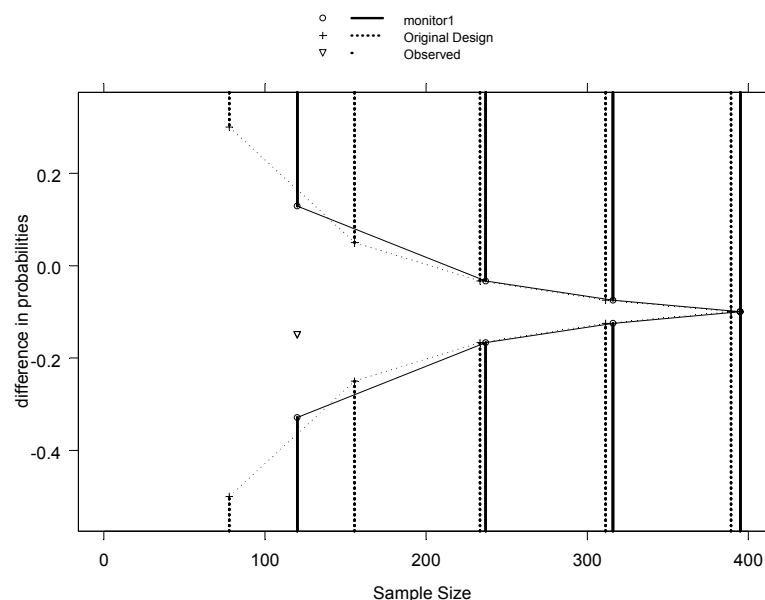


Figure 7.4: Display of the modified stopping boundaries (solid lines) for the first interim analysis, the original design (broken lines), and the current crude estimate of treatment effect.

From the command line, the first interim analysis can be performed by the commands

```
> monitor1 <- seqMonitor(tutorial.obf, response1,
+   treatment1)
> plot(monitor1)
```

We thus simulate the continuation of the clinical trial by simulating additional data. We consider the setting in which the next analysis is performed after an additional 100 subjects (50 per arm) are accrued to the study. The following S+ code simulates this additional data and stores it as a dataset named **secondData**.

```
> treatment2 <- rep(0:1, each=50)
> response2 <- rbinom(100, 1, ifelse(treatment2==1, .35, .50))
> treatment2 <- c(treatment1, treatment2)
> response2 <- c(response1, response2)
> secondData <- data.frame(response2, treatment2)
```

In order to perform the second interim analysis, we complete the **Group Sequential Monitoring** dialog as shown in Figure 7.5.

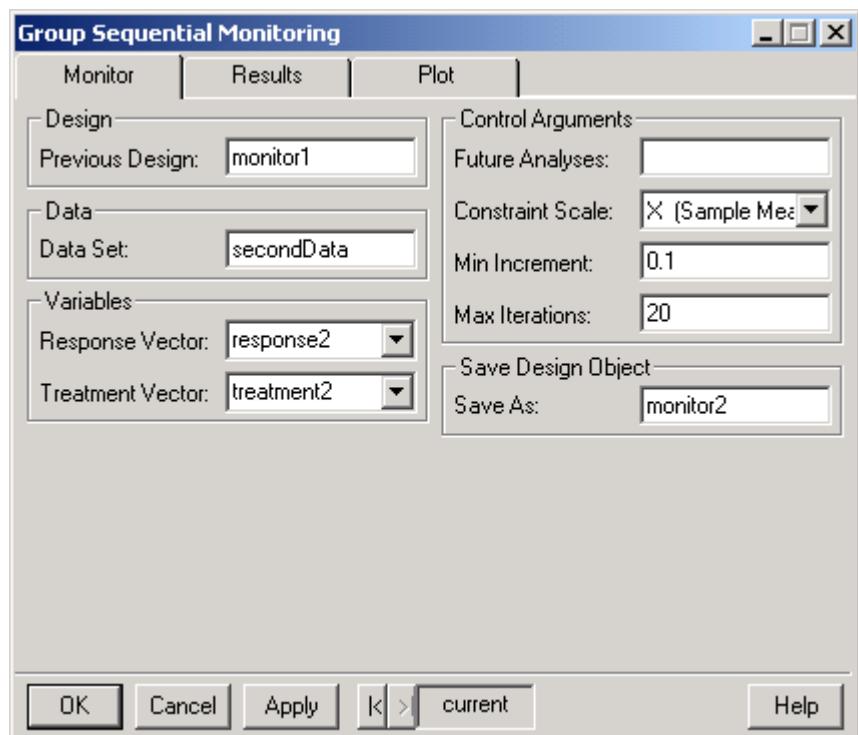


Figure 7.5: The completed dialog for the second interim analysis of the simulated data.

From the command line, the second interim analysis can be performed by the commands

```
> monitor2 <- seqMonitor(monitor1, response2,  
+ treatment2)
```

Note that the second interim analysis uses the information stored in the **seqMonitor** object **monitor1**, which was created at the first interim analysis. The data is taken from dataset **secondData**, and the results of the second interim analysis are to be stored in a new **seqMonitor** object named **monitor2**. After clicking **Apply**, the following output appears in the **Report** window.

```
*** Sequential Monitor ***  
Call:  
seqMonitor(x = monitor1, response = secondData$response2, treatment  
= secondData$treatment2, constraint.scale = "X", min.increment =  
0.1, maxiter = 20)  
  
RECOMMENDATION:  
Stop with decision for Lower Alternative Hypothesis  
  
OBSERVED STATISTICS:  
Sample Size Crude Estimate Z Statistic  
120 -0.1500 -1.664  
220 -0.2091 -3.175  
  
INFERENCE:  
analysis.index observed MLE BAM RBadj  
1 2 -0.2091 -0.2091 -0.1954 -0.2057  
  
Inferences based on Analysis Time Ordering:  
MUE P-value **** CI ****  
1 -0.2086 0.000823 (-0.3379, -0.0791)  
  
Inferences based on Mean Ordering:  
MUE P-value **** CI ****  
1 -0.2051 0.0008317 (-0.3339, -0.0781)  
  
MONITORING BOUNDS:  
Call:  
"(not shown, is (your seqMonitor object)$seqDesignCall)"  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable
```

```

Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
  Null hypothesis : Theta >= 0           (size = 0.025)
  Alternative hypothesis : Theta <= -0.1999   (power = 0.975)

```

STOPPING BOUNDARIES: Sample Mean scale			
	a	d	
Time 1 (N= 120)	-0.3290	0.1291	
Time 2 (N= 220)	-0.1753	-0.0245	
Time 3 (N= 309)	-0.1248	-0.0750	
Time 4 (N= 386)	-0.0999	-0.0999	

From the above output, you can see that the observed difference in 28 day mortality at the second analysis is -0.2091. According to the modified stopping rule, early termination of the study with a decision for efficacy would only occur at the second analysis if that crude estimate of treatment effect were less than -0.1753. Because the simulated results exceeded that boundary, the output provides the recommendation to stop with a decision for the lower alternative hypothesis.

Also provided is correct frequentist inference adjusted for the stopping rule actually used in the monitoring procedure. There are several methods for computing adjusted point estimates, confidence intervals, and *p*-values returned in the output. Emerson & Fleming (1990) generally recommended the use of the bias adjusted mean for the point estimate of treatment effect and the use of the confidence interval and *p*-value based on the sample mean ordering. We would thus report an estimated improvement in the proportion of patients surviving 28 days of 0.195 (95% CI .078 to .334, one-sided *P* = 0.0008).

Constrained Boundaries on the Error Spending Scale

We can also examine how the monitoring of this trial would proceed using the error spending approach of Lan & DeMets (1983). For this example, we do not allow the maximal sample size to be changed, and we specify that the stopping boundaries for previously conducted interim analyses be constrained on the error spending function scale.

In order to perform the first interim analysis, we

- Fill in the **Previous Design** text box with the name of the **seqDesign** object containing the clinical trial design: **tutorial.obf**.

- Fill in the **Data Set** text box with the name of the data set containing the clinical trial data: **firstData**.
- Select the name of the vector recording the indicators of 28 day mortality for each patient: **response1**.
- Select the name of the vector indicating the treatment: **treatment1**.
- Specify that future analyses should be conducted at the sample sizes planned at the time of study design by entering **156, 234, 311, 389** in the **Future Analyses** text box.
- Specify that stopping boundaries at previously conducted interim analyses are to be constrained on the error spending scale by selecting **E (Error Spending)** from the **Constraint Scale** pulldown menu.
- fill in a name to be used to store the **seqMonitor** object which results from this interim analysis: **emonitor1**.

The completed **Monitor** tab of the **Group Sequential Monitoring** dialog appears as shown in Figure 7.6.

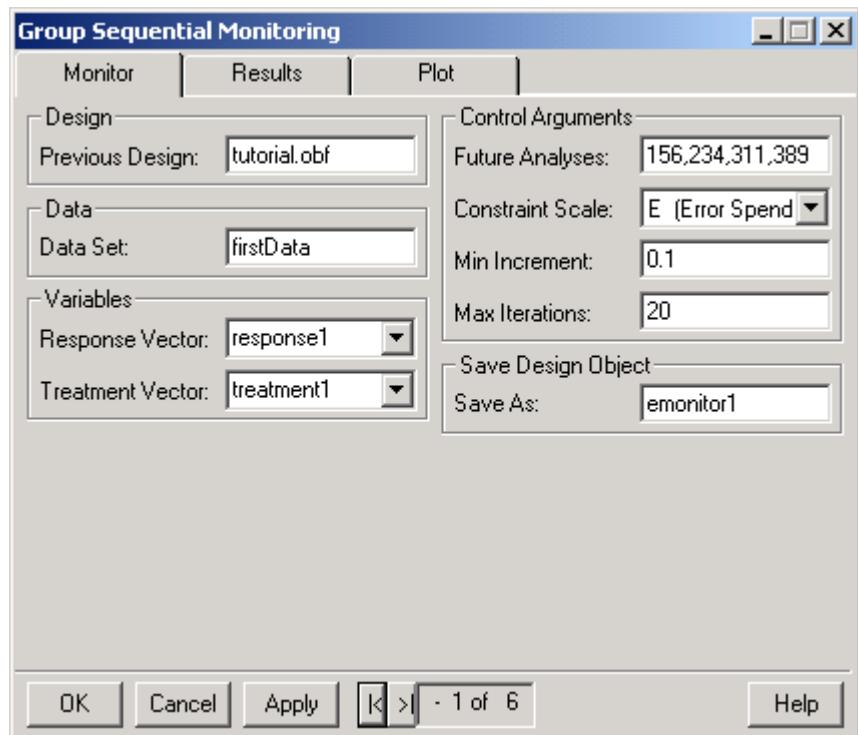


Figure 7.6: The completed dialog when using the error spending approach for the first interim analysis of the simulated data using *tutorial.obf*.

After clicking **OK** the following output appears in the **Report** window.

```
*** Sequential Monitor ***
Call:
seqMonitor(x = tutorial.obf, response = firstData$response1,
            treatment = firstData$treatment1, future.analyses = c(156, 234,
            311, 389), constraint.scale = "E", min.increment = 0.1, maxiter =
            20)

RECOMMENDATION:
  Continue

OBSERVED STATISTICS:
  Sample Size Crude Estimate Z Statistic
    120           -0.15        -1.664

MONITORING BOUNDS:
```

```
Call:  
"not shown, is (your seqMonitor object)$seqDesignCall)"  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
One-sided hypothesis test of a lesser alternative:  
Null hypothesis : Theta >= 0 (size = 0.025)  
Alternative hypothesis : Theta <= -0.2016 (power = 0.975)  
  
STOPPING BOUNDARIES: Sample Mean scale  
a d  
Time 1 (N= 120) -0.3024 0.1008  
Time 2 (N= 234) -0.1672 -0.0345  
Time 3 (N= 311) -0.1261 -0.0755  
Time 4 (N= 389) -0.1008 -0.1008
```

From the command line, the first interim analysis can be performed by the commands

```
> emonitor1 <- seqMonitor(tutorial.obf, response1,  
+ treatment1,  
+ future.analyses=c(156, 234, 311, 389),  
+ constraint.scale="E")  
> plot(emonitor1)
```

From the above output, we again see that the observed difference in 28 day mortality is -0.15. According to the modified stopping rule, early termination of the study with a decision for efficacy would only occur if that crude estimate of treatment effect were less than -0.3024, and early termination with a decision for futility would only occur if the crude estimate of treatment effect were greater than 0.1008. Hence, the output provides the recommendation to continue.

The above stopping boundary differs from that obtained using the unified family approach on the sample mean scale. When implementing the O'Brien-Fleming stopping rule on the error spending function scale, S+SEQTRIAL interpolates the error spending function boundaries from **tutorial.obf**. (You can see the boundaries displayed on this scale by choosing the **Display Scale** on the **Reports** tab.) An O'Brien-Fleming design for the specified schedule of analyses and type I error of 0.025 corresponds to spending a type I error of 0.00013 at the first analysis having 120 accrued subjects.

However, the error spending implementation used interpolation of the error spending function from the **tutorial.obf** error spending function, which spent 0.0000034 at a sample size of 78 and 0.00073 at a sample size of 156. Hence, the error spending implementation resulted in a design which spends a type I error of 0.00040 at a sample size of 120.

In order to perform the second interim analysis, we complete the **Group Sequential Monitoring** dialog as shown in Figure 7.7.

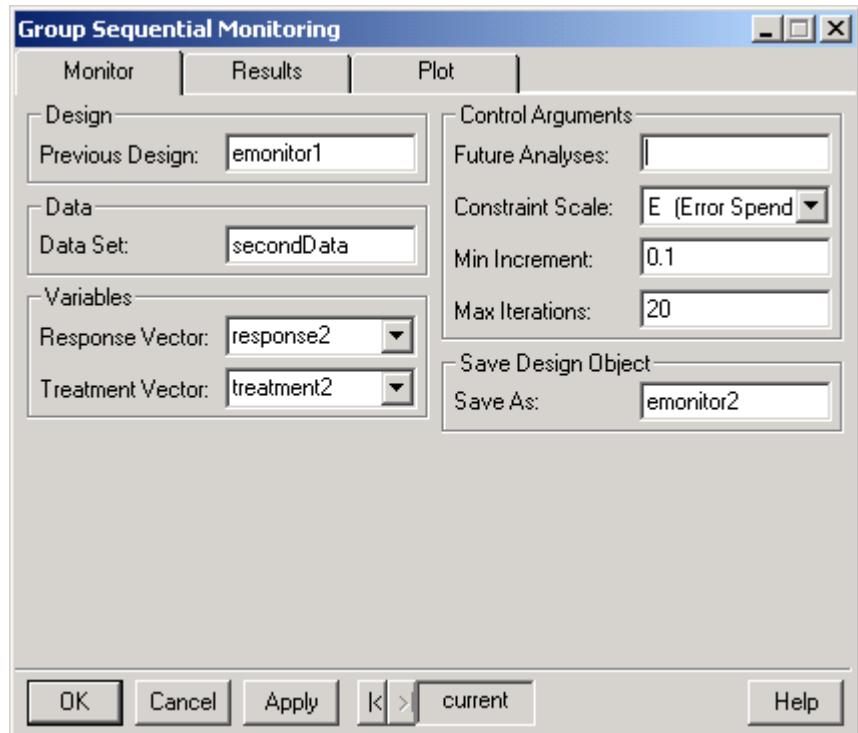


Figure 7.7: The completed dialog when using the error spending approach for the second interim analysis of the simulated data.

Note that the second interim analysis uses the information stored in the **seqMonitor** object **emonitor1**, which was created at the first interim analysis. The data is taken from dataset **secondData**, and the results of the second interim analysis are to be stored in a new **seqMonitor** object named **emonitor2**.

From the command line, the second interim analysis can be performed by the commands

```
> emonitor2 <- seqMonitor(emonitor1, response2,
+      treatment2)
```

After clicking **OK**, the following output appears in the **Report** window.

```
*** Sequential Monitor ***
Call:
seqMonitor(x = emonitor1, response = secondData$response2,
treatment = secondData$treatment2, future.analyses = c(311, 389),
constraint.scale = "E", min.increment = 0.1, maxiter = 20)

RECOMMENDATION:
Stop with decision for Lower Alternative Hypothesis

OBSERVED STATISTICS:
Sample Size Crude Estimate Z Statistic
    120        -0.1500       -1.664
    220        -0.2091       -3.175

INFERENCE:
analysis.index observed      MLE      BAM     RBadj
1                  2 -0.2091 -0.2091 -0.1926 -0.2007

Inferences based on Analysis Time Ordering:
      MUE   P-value      **** CI ****
1 -0.2075 0.001054 (-0.3372, -0.0766)

Inferences based on Mean Ordering:
      MUE   P-value      **** CI ****
1 -0.2041 0.001062 (-0.3333, -0.0758)

MONITORING BOUNDS:
Call:
"(not shown, is (your seqMonitor object)$seqDesignCall)"
```

PROBABILITY MODEL and HYPOTHESES:

```

Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
    Null hypothesis : Theta >= 0           (size = 0.025)
    Alternative hypothesis : Theta <= -0.1994   (power = 0.975)

```

STOPPING BOUNDARIES: Sample Mean scale

	a	d
Time 1 (N= 120)	-0.2991	0.0997
Time 2 (N= 220)	-0.1746	-0.0249
Time 3 (N= 311)	-0.1245	-0.0750
Time 4 (N= 389)	-0.0997	-0.0997

From the above output, we can see that the observed difference in 28 day mortality at the second analysis is -0.2091. According to the modified stopping rule, early termination of the study with a decision for efficacy would only occur at the second analysis if that crude estimate of treatment effect were less than -0.1746. Because the simulated results exceeded that boundary, the output provides the recommendation to stop with a decision for the lower alternative hypothesis.

Also provided is correct frequentist inference adjusted for the stopping rule actually used in the monitoring procedure. There are several methods for computing adjusted point estimates, confidence intervals, and *p*-values returned in the output. Emerson & Fleming (1990) generally recommended the use of the bias adjusted mean for the point estimate of treatment effect and the use of the confidence interval and *p*-value based on the sample mean ordering. We would thus report an estimated improvement in the proportion of patients surviving 28 days of 0.193 (95% CI .076 to .333, one-sided *P* = 0.0011).

The above inference differs only slightly from that obtained when the stopping rule was implemented by constraining boundaries on the sample mean scale. The reason for the discrepancy is that different implementations of the stopping rule result in slightly different sampling schemes. Frequentist inference is affected by the exact sampling scheme used, thus the inference should be different. The fact that the difference is so slight is testimony to the similarity between the various approaches in this simulated case.

You can also see the effect of needing to estimate the variability of the test statistic on the correspondence between the error spending scale and the sample mean scale at previously conducted analyses. When

using the error spending scale, the sample mean critical value at the first analysis was -.3024, which, when using the estimated variance at the first analysis, corresponded to spending a type I error of 0.0004. At the second analysis, the sample mean critical value for the first analysis was reported as -.2991, which corresponds to spending a type I error of 0.0004 when using the estimated variance at the second analysis.

Additional Examples

In the above examples, we maintained statistical power to detect the alternative when we used constrained boundaries on the sample mean scale, and we maintained the maximal sample size when we used constrained boundaries on the error spending scale. We could have done either of these differently. The decisions about maintaining power versus maximal sample size and about constraining the boundaries on any particular scale can be made independently.

A further detailed example of monitoring is included as part of *Case Study #2: Unplanned Analyses* (see page 261). Additional examples are also available in the Help file for seqMonitor. Type:

```
> ?seqMonitor
```

SPECIFYING A MONITORING PROCEDURE

When monitoring a clinical trial, S+SEQTRIAL performs an analysis of the clinical trial data accrued to date according to the probability model specified in the original design. Using the information regarding the number and timing of all previous interim analyses along with the sample size at the current analysis, the function calculates the stopping boundary appropriate for use at the current analysis. This stopping boundary is computed according to the design family parameters specified at the design stage, but constrained to agree with the boundaries actually used at all prior interim analyses.

Modified boundaries are computed so that the Type I statistical error is maintained at the level specified during the design stage. In addition, you can decide whether to maintain the maximal sample size or the power of the study. If the maximal sample size is not constrained, S+SEQTRIAL computes an estimate of the maximal sample size which maintains the statistical power to detect the design alternative as specified at the design stage. This estimate of the maximal sample size is based on a schedule of monitoring which reflects all prior interim analyses, and an estimate of the number and timing of future analyses; by default the number and relative timing are extract from the most current design (the original design, or update at the most recent interim analysis).

The exact way in which information about the desired monitoring plan should be entered into the **Group Sequential Monitoring** dialog is described below.

Previous Design

At each interim analysis, the stopping rule parameters chosen at the time of study design are used to compute the stopping boundaries appropriate for the current sample size. At the first interim analysis, those stopping rule parameters are taken from the **seqDesign** object created during the planning of the study. At all later analyses, the stopping rule parameters, as well as the stopping boundaries actually used at all previously conducted interim analyses, are taken from the **seqMonitor** object created at the immediately preceding interim analysis. You indicate the appropriate source of the stopping rule parameters by entering the name of the **seqDesign** or **seqMonitor** object in the **Previous Design** text box.

Data

When monitoring a clinical trial, S+SEQTRIAL performs the data analysis using data stored in a data frame. The data frame must contain one variable measuring the response appropriate for the probability model, and another variable indicating the treatment group. The name of the data frame is entered in the **Data Set** text box, and then the response and treatment variables are selected from the pulldown menu in the **Response Vector** and **Treatment Vector** fields, respectively.

Future Analyses

In order to provide the best correspondence between the operating characteristics for the actual stopping rule used when monitoring the data and those for the original stopping rule selected at the time of study design, it is useful to update the planned schedule of analyses to reflect the best estimates of the number and timing of future analyses.

By default, S+SEQTRIAL uses a modification of the schedule estimated at the time of study design. In this modification, the schedule of analyses agrees exactly with the sample sizes at which previously conducted interim analyses were performed, and uses the sample size determined from the current data to register the timing of the current analysis, dropping all analyses which were planned at smaller sample sizes but not performed. Then, if the current analysis is within a minimal increment (specified in the **Min. Increment** text box and measured as a proportion of the maximal sample size) of a planned future analysis, it is assumed that the current analysis replaces that future analysis. Otherwise, future analyses are estimated to occur according to the proportion of maximal information specified in the original trial design.

When you do not want the maximal sample size recomputed to account for changes in statistical power due to poor estimates of response variability or number of timing and analyses, you need only supply the sample sizes at which you anticipate performing future analyses. For instance, in the example presented in the previous section, you could prevent re-computation of maximal sample size by entering **156, 234, 311, 389** in the text box for **Future Analyses**. In this way you signify to S+SEQTRIAL that you estimate that all interim analyses after the first adhere to the schedule originally planned.

Constraint Scale

The common feature to all flexible monitoring procedures implemented in S+SEQTRIAL is the use of constrained boundaries. That is, when performing the j th interim analysis, S+SEQTRIAL computes the current and future critical values according to the stopping rule parameters supplied in the clinical trial design. However, because all previous analyses have already been performed, it is imperative that the stopping boundaries computed at the j th analysis agree exactly with the critical values used at the previous analyses.

There are a number of boundary scales that can be used to express a stopping boundary. As noted in Chapter 5 on page 112, there is a 1:1 correspondence between the various boundary scales when designing a clinical trial, and thus no great distinction is made according to which boundary scale is used to define a stopping rule. However, when implementing a stopping rule, it is generally the case that some estimate of the variance must be used to compute the test statistic. This causes the implementation of a stopping rule to vary slightly when using different boundary scales. For that reason, S+SEQTRIAL allows you to specify the boundary scale to be used when constraining the boundaries to agree with previously performed interim analyses. The boundaries can be constrained on any of the valid monitoring scales: Sample Mean, Z Statistic, Fixed Sample P value, or Error Spending scale, except that only Error Spending may be used if the original design family was the error spending family.

For the error spending scale, the stopping rule specified at the design stage is converted to the error spending scale and then used as the basis for an error spending function, with interpolation used to define the error spending function at analysis times not specified at the design stage. This use of linear interpolation means that monitoring rules which constrain on the error spending scale may not agree well with the operating characteristics of the original design if that original design was defined on one of the other scales.

Constraining boundaries on the error spending scale is exactly equivalent to using the error spending function approach of Lan & DeMets (1983) (if maximal sample size is maintained at the value chosen during study design) or Pampallona, Tsiatis, & Kim (1995) (if the statistical to detect the alternative is maintained at the predefined level used in clinical trial design).

**Minimum
Spacing of
Analyses**

If the current sample size is very close to that of an estimated analysis time, S+SEQTRIAL assumes that the current analysis takes the place of that estimated analysis time. You can control the definition of the minimal spacing between the current analysis and the next estimated analysis through the text box **Min. Increment**. Values entered in this text box are measured as a proportion of the maximal sample size. Hence a value of 0.1 corresponds to the decision to not perform any two interim analyses if the number of accrued subjects between those analyses would be less than 10% of the maximal sample size.

It should be noted that this field is only used when S+SEQTRIAL is estimating the future schedule of analyses. You may, of course, later perform an interim analysis at times other than those estimated in your original design or estimated when modifying the stopping rule. The methods used by S+SEQTRIAL to update the stopping rule maintain the type I error at the desired level so long as the decisions regarding the timing of the interim analyses are independent of the observed estimates of treatment effect.

**Maximum
Iterations**

When S+SEQTRIAL updates the maximal sample size to maintain the statistical power to detect the alternative, the computation of the modified stopping rule often involves an iterative search. The text box **Max Iterations** allows you to control the number of iterations used to find the modified stopping rule. As a general rule, the default value of 20 should be more than adequate.

EVALUATING A MONITORING RULE

After performing an interim analysis, S+SEQTRIAL creates a **seqMonitor** object which stores the results of the data analysis, the modified stopping rule, and the recommended decision regarding early termination of the clinical trial. The **seqMonitor** object inherits many properties of a **seqDesign** object. Hence, all the evaluative procedures that are used when initially planning a clinical trial can also be used on the monitoring results. That is, plots of stopping boundaries, potential frequentist inference, ASN curves, power curves, and stopping probabilities are easily obtained using the same functions as used for design objects. Hence, the **Results** and **Plots** tabs of the **Group Sequential Monitoring** dialog provide nearly identical capabilities as those tabs in the dialogs used to create designs. Such capability is often important when situations external to the clinical trial motivate a re-examination of the clinical trial design as discussed in *Case Study #2: Unplanned Analyses* (see page 261)..

REPORTING RESULTS

8

Overview	230
Adjusted Point Estimators	231
Bias Adjusted Mean	231
Rao-Blackwell Adjusted Estimate	231
Median Unbiased Estimate	231
Comparing Estimators	232
Adjusted Confidence Intervals	233
Sample Mean Ordering	233
Analysis Time Ordering	233
Comparing Orderings	233
Adjusted P-Values	235
Obtaining Adjusted Estimates and P-Values	236

OVERVIEW

At the end of a clinical trial, the results of an analysis are typically summarized by providing:

- a point estimate (best estimate) of the treatment effect;
- a confidence interval for the treatment effect;
- a p -value for the test of the null hypothesis that the treatment effect is zero.

Group sequential testing does not alter the importance of reporting such statistics at the end of a trial. The stopping rule does, however, greatly alter the formulas and methods used to compute these statistics.

The purpose of a group sequential stopping rule is to terminate a study whenever the observed results are so extreme as to suggest an ethical need or advantage in efficiency. By preferentially stopping the study when such extreme results are observed, however, some bias is introduced into the usual point estimators of the treatment effect. For example, in the normal and binomial proportions models, the maximum likelihood estimates of the treatment effect are unbiased and efficient in the absence of interim analyses. However, once interim analyses are performed, the maximum likelihood estimates are biased and have larger variability than other adjusted estimates. Furthermore, the p -values computed by most statistical analysis routines are incorrect for the group sequential setting, and confidence intervals computed using methods appropriate for fixed sample tests have incorrect coverage probabilities (Whitehead, 1986; Emerson & Fleming, 1990).

This chapter describes how to report your results while adjusting for the existence of a stopping rule.

ADJUSTED POINT ESTIMATORS

By preferentially stopping a study when extreme results are observed, bias is introduced into the usual point estimators of the treatment effect. Therefore, a number of revised estimators of the treatment effect have been investigated for use in reporting the results of group sequential trials. S+SEQTRIAL provides three such estimators, discussed below.

For completeness, S+SEQTRIAL also includes the maximum likelihood estimate. This estimator represents the naive estimate of the treatment effects that would have been appropriate, and in some sense optimal, if no interim analyses had been performed. But because it makes no adjustment for the stopping rule, it gives a biased estimate of treatment effect in a group sequential trial.

Bias Adjusted Mean

The bias adjusted mean estimator (Whitehead, 1986) is based on the method of moments; the estimator is the parameter value (estimated by an iterative search) for which the mean of the true group sequential sampling distribution equals the observed statistic.

Rao-Blackwell Adjusted Estimate

The Rao-Blackwell adjusted estimate (Emerson & Fleming, 1990; Emerson, 1993, Emerson & Kittelson, 1997) uses the idea of the Rao-Blackwell improvement theorem to reduce the variability of an unbiased estimator of the treatment effect in the normal probability model. The resulting estimator has minimum variance among all estimates based solely on information from analyses that have been already been performed. Thus, in some sense, this estimator is like a uniform minimum variance unbiased estimate (UMVUE) for the normal model. This estimator can also be used asymptotically in other probability models. Though it is not strictly unbiased in those other settings, the bias is greatly reduced.

Median Unbiased Estimate

The median unbiased estimate (Whitehead, 1997; Emerson & Fleming, 1990) is based on matching quantiles; the estimator is the parameter value (estimated by an iterative search) for which the median of the true group sequential sampling distribution equals the observed statistic.

This estimate depends on defining an ordering of the possible outcomes from the clinical trial. Two such orderings are implemented in S+SEQTRIAL: ordering on the sample mean (the estimate of treatment effect) (Emerson & Fleming, 1990), and ordering on the analysis time (Jennison & Turnbull, 1984; Tsiatis, Rosner, & Mehta, 1984). See page 233 for more information.

Comparing Estimators

Emerson & Fleming (1990) compared the behavior of the three adjusted estimators of treatment effect described above. They reported that the bias adjusted mean tends to have lower mean squared error than the Rao-Blackwell estimator over a range of interesting cases. It is more closely related to the sample mean ordering of the outcome space, and thus tends to agree more closely with the recommended methods of computing p -values and confidence intervals. The bias adjusted mean does have the disadvantage of depending upon the number and timing of future analyses, although Emerson & Fleming (1990) found that such dependence was largely inconsequential.

In the normal probability model, the Rao-Blackwell estimator is, of course, unbiased, and it has the advantage of not requiring information about the number and timing of future analyses. It can, however, have a higher mean squared error than the bias adjusted mean. It can also happen that this estimator is not included in the confidence intervals based on the sample mean ordering, though this is a rare event. In probability models other than the normal and binomial proportions, this estimator is not totally unbiased due to the logarithmic link function used in those other models.

The median unbiased estimate generally has greater bias and variability than either the bias adjusted mean or the Rao-Blackwell adjusted estimate. A median unbiased estimate based on the sample mean ordering tends to have less bias and variability than an estimate based on the analysis time ordering. The sample mean ordering does have the drawback that it depends on the number and timing of future analyses.

ADJUSTED CONFIDENCE INTERVALS

In order to compute a confidence interval which adjusts for the stopping rule used in a group sequential trial, an ordering of possible clinical trial outcomes must be chosen. There is no uniformly optimal choice for such an ordering.

In group sequential testing, the issue is how to treat outcomes observed at different analyses (see Emerson & Fleming, 1990). S+SEQTRIAL offers two approaches.

Sample Mean Ordering

In this approach (Emerson & Fleming, 1990), one result is judged more extreme than another result according to whether the maximum likelihood estimate of the treatment effect is more extreme. Thus, a treatment effect measured by a hazard ratio of 1.5 is higher than a treatment effect measured by a hazard ratio of 1.4, regardless of the analysis time.

Analysis Time Ordering

In this approach (Tsiatis, Rosner, & Mehta, 1984), results that led to earlier termination of the study are judged to be more extreme than those observed at later analyses. Results that exceed an upper boundary for the treatment effect at a specific analysis are higher than all results exceeding the upper boundary at later analyses, and also higher than all results less than the lower boundary at any analysis. Thus, a treatment effect measured by a hazard ratio of 1.5, which was judged so high as to warrant early termination of the study, is less than a hazard ratio of 1.4 which was similarly judged high enough to warrant termination of the study at an earlier analysis. This definition assumes a study design in which there are no inner ("b" or "c") stopping boundaries at any interim analysis. This ordering is not defined for two-sided group sequential tests that allow early stopping under both the null and alternative analyses.

Comparing Orderings

Emerson & Fleming (1990) investigated the relative behavior of the sample mean and analysis time orderings with respect to the average width of confidence intervals. The sample mean ordering tends to average shorter confidence interval lengths for the same coverage probabilities. Because it is also defined for all group sequential test designs, it is the recommended method of computing confidence

intervals. Unlike the analysis time ordering, however, the sample mean ordering does depend on the number and timing of future analyses, but such dependence was found to be fairly slight by Emerson & Fleming (1990).

ADJUSTED P-VALUES

Standard statistical software used to analyze clinical trial data assumes that data were collected in a fixed sample study with no possibility of early termination. Thus, the p -values reported in the statistical output assume an incorrect sampling distribution for a group sequential test statistic. S+SEQTRIAL allows the computation of the true group sequential p -value.

Note: The adjusted p -value is typically different than the p -value returned when you display group sequential boundaries on the Fixed-Sample P scale.

As with the calculation of confidence intervals (see page 233), the definition of an adjusted p -value depends upon the ordering of the possible outcomes for the clinical trial. Again, the sample mean based ordering is recommended, due to its better behavior. You may, however, also request a p -value based on the analysis time ordering for those group sequential stopping rules for which it is defined.

In general, there is very little substantive difference between the p -values derived for the sample mean and analysis time orderings when those p -values would correspond to statistically significant results.

OBTAINING ADJUSTED ESTIMATES AND P-VALUES

When the S+SEQTRIAL **Group Sequential Monitoring** dialog (or command line function seqMonitor) is used to monitor a clinical trial (see Chapter 7), adjusted point estimates, *p*-values, and 95% confidence intervals are automatically returned whenever an interim analysis of the data suggests early termination of the study, or when the final analysis of the data is conducted.

You can also examine inferences corresponding to observations on the stopping boundaries using the **Summary Tables** option on the **Results** tab. For each analysis time, the stopping boundaries are first reported, then four different point estimates are given (see page 231 for more information on these estimators):

- MLE, maximum likelihood estimate based on the observed statistics;
- BAM, bias adjusted mean estimate;
- RBadj, Rao-Blackwell adjusted estimate (under some conditions this is a UMVUE, uniformly minimum-variance unbiased estimate);
- MUE, median unbiased estimate.

Finally, the *p*-values and 95% confidence intervals are reported. For example, here are the inferences at the boundaries for the design tutorial.obf produced in Chapter 3.

		Inferences at the Boundaries	
		*** a Boundary ***	*** d Boundary ***
Time 1	Boundary	-0.500	0.300
	MLE	-0.500	0.300
	BAM	-0.478	0.278
	RBadj	-0.500	0.300
	MUE	-0.472	0.272
	P-value	0.000	0.997
	95% Conf Int	(-0.639, -0.280)	(0.080, 0.439)
Time 2	Boundary	-0.250	0.050
	MLE	-0.250	0.050
	BAM	-0.234	0.034
	RBadj	-0.250	0.050
	MUE	-0.233	0.033
	P-value	0.001	0.682
	95% Conf Int	(-0.365, -0.093)	(-0.107, 0.165)

Time 3	Boundary	-0.167	-0.033
	MLE	-0.167	-0.033
	BAM	-0.155	-0.045
	RBadj	-0.163	-0.037
	MUE	-0.154	-0.046
	P-value	0.005	0.212
	95% Conf Int	(-0.266, -0.037)	(-0.163, 0.066)
Time 4	Boundary	-0.125	-0.075
	MLE	-0.125	-0.075
	BAM	-0.120	-0.080
	RBadj	-0.118	-0.082
	MUE	-0.115	-0.085
	P-value	0.015	0.049
	95% Conf Int	(-0.216, -0.011)	(-0.189, 0.016)
Time 5	Boundary	-0.100	-0.100
	MLE	-0.100	-0.100
	BAM	-0.100	-0.100
	RBadj	-0.100	-0.100
	MUE	-0.100	-0.100
	P-value	0.025	0.025
	95% Conf Int	(-0.200, 0.000)	(-0.200, 0.000)

From the command line, you can print the same information using the `summary` function. For example:

```
> summary(tutorial.obf)
```

You can also compute the values directly using the `seqInference` function. With no arguments supplied but a group sequential design object, the `seqInference` function returns the adjusted estimates and *p*-values corresponding to observations on the stopping boundaries. If you supply an analysis time (`analysis.index`) and an observed statistic (`observed`), the `seqInference` function returns inferences based on the ordering of possible outcomes specified by the `ordering` argument.

You can plot the inferences for each boundary at each analysis using the **Inference** option on the **Plot** tab. For example, Figure 8.1 shows point estimates (BAM) and 95% confidence intervals for design `tutorial.obf`.

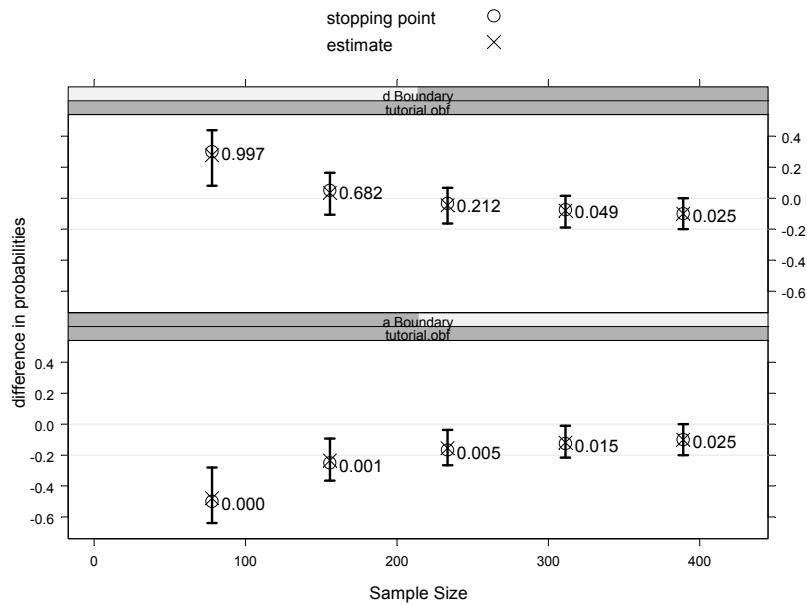


Figure 8.1: Point estimates (BAM) and 95% confidence intervals for the design tutorial.obf.

To plot point estimates other than the bias-adjusted mean (BAM), you must use the seqPlotInference function from the command line, as described below.

From the command line, you can plot the point estimates and 95% confidence intervals using the seqPlotInference function. For example:

```
> seqPlotInference(tutorial.obf)
```

The default point estimate is the bias-adjusted mean estimate (BAM). Use the estimate argument to the seqPlotInference function to plot a different estimate. For example:

```
> seqPlotInference(tutorial.obf, estimate="MLE")
```

CASE STUDY #1: HYBRID TESTS

9

Overview	240
Limitations of the Standard Designs	242
Selecting the Model	242
Specifying the Designs	242
Evaluating the Designs	247
Hybrid Tests	253
Using the Epsilon Shift Parameter	253
Evaluating the Designs	257

OVERVIEW

Left untreated, tumors growing close to the spine can lead to a variety of neurologic symptoms, paralysis, or death. Treatment of such tumors poses special problems, however, because standard radiation therapy is not precise enough to avoid radiation injury to the spinal cord. In this case study, a special stereotactic spinal radiosurgery (SSR) frame was developed that would establish a precise coordinate system to be used as a reference in directing the radiation beam.

The new SSR frame had been tested on a compassionate therapy basis in a few patients at the time the planning for this clinical trial was taking place. As more involved testing was being considered, a concern was raised about the need for a concurrent control group. Although it is not unusual that early trials of cancer therapies are uncontrolled, the disadvantages of such a single arm trial seemed greater for this setting. The trial of the SSR frame was to be undertaken in patients who were generally terminally ill and for whom other avenues of treatment were exhausted. However, use of the SSR frame requires a patient to undergo up to 12 hours of general anesthesia. Hence, it was not immediately clear what the natural course of untreated disease should be in these patients. The potential toxicities from excessive radiation exposure might present clinically in much the same way that the progression of disease presented. Hence, it was proposed that a controlled clinical trial be conducted with sequential monitoring to ensure that patients would not be needlessly exposed to inferior treatments. The primary endpoint for monitoring of the trial was based on a composite measure of patient death and radiation damage to the spinal cord, as defined by a 50% decrease in a quantitative measure of neurologic function.

Note that this primary endpoint was directed at patient safety. In fact, the presumed benefit of the treatment was not so much curative as palliative. The researchers hypothesized that patients using SSR would have better pain relief and other measures of quality of life, with no decrease in survival. Because the true goal of the study was to detect improvements in quality of life, the role of a stopping rule is to establish approximate equivalence between the SSR and control group with respect to death and radiation injury, although of course the researchers would be interested in showing an improvement in survival with SSR. In any case, if the two treatment arms were

sufficiently equivalent with respect to survival and radiation toxicity, the study would be analyzed for differences with respect to quality of life.

The first group sequential trial designs proposed resulted in stopping boundaries corresponding to differences in adverse outcomes between the treatment arms with which the researchers were uncomfortable. The question is how to choose a design that does not exceed the differences in mortality that the researchers could subjectively tolerate, but still affords the opportunity to examine benefits with respect to primary or secondary outcomes. This case study demonstrates that when the standard group sequential designs do not satisfy these requirements, it is still possible to find a satisfactory design using the expanded “hybrid” tests supported by S+SEQTRIAL.

LIMITATIONS OF THE STANDARD DESIGNS

Let's first demonstrate the limits to what can be achieved by the standard designs in this case. In the next section, we'll overcome these limitations by exploring an expanded family of designs available in S+SEQTRIAL.

Selecting the Model

The proposed study has a binary endpoint measuring toxicity, with subjects assigned with equal probability to a treatment group and to a placebo control group. Hence, the model is a two-sample Binomial proportions hypothesis test. Choose **SeqTrial ► Two Samples ► Binomial Proportions** to launch the appropriate S+SEQTRIAL dialog for this trial.

All of the commands needed to run this case study from the command line are contained in a help file. Type:

```
> help(case1.seqtrial)
```

Specifying the Designs

The first design uses an O'Brien-Fleming boundary relationship ($P = 1$) for the upper and lower boundaries. The researchers predicted that a total of 200 patients would be available over a four year period, and interim analyses would be conducted on a yearly basis. Because the overall goal of the study was to examine quality of life issues, there was no facility for early stopping of the study with a failure to reject the null hypothesis.

1. On the **Design** tab, select the **Min. Difference** option for the computational task. In this case, S+SEQTRIAL computes the alternative for which a given sample size has the desired power; you must specify the sample size and the power.
2. Specify the **Probabilities**. Set the **Significance Level** to .05 and the **Power** to .95, which is in keeping with FDA recommendations for a two-sided test.
3. The design involved a total of 200 patients, 100 on each treatment arm. Set the **N** field in the **Sample Sizes** groupbox to 200. Ensure that the **Ratio** field is set to 1 (the default),

indicating a 1:1 ratio of the sample sizes between the treatment and the control groups, and hence 100 patients in each group.

4. Under the null hypothesis, the assumed toxicity rates are assumed to be the same. The baseline toxicity rate was taken to be .5 based on a rough subjective prior. Ensure that the **Null Proportions** field is set to **0.5, 0.5** (the default).
5. Set the **Variance Method** to **null**, and the **Test Type** to **two.sided**.
6. Set the **No. of Analyses** to 4. The analyses are evenly spaced according to sample size—in this case, after 50, 100, 150, and 200 subjects are accumulated.
7. Set the **Null Bnd. Shape** field to **No Early Stopping**, indicating that you do not ever want to stop the study early with a decision for the null hypothesis (a failure to reject the null hypothesis). Because the overall goal of the study was to examine quality of life issues, there was no facility for early stopping of the study with a failure to reject the null hypothesis.
8. Ensure that the **Alt. Bnd. Shape** is set to **0br-F1 (P=1)** (the default), which corresponds to O'Brien-Fleming boundary relationships.
9. Save the design object under the name **ssr.ofb** using the **Save As** field.

The **Design** tab should now look like that shown in Figure 9.1.

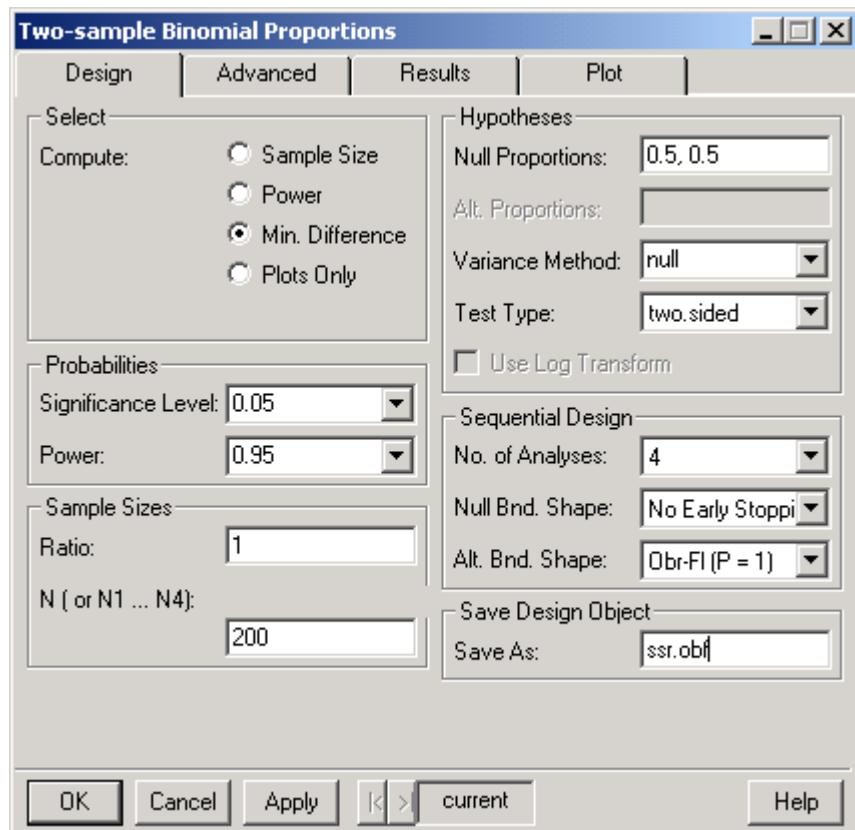


Figure 9.1: The Design tab for design *ssr.obf*.

In this study, the researchers had concerns about the excess number of adverse events (radiation toxicity or death) that might be observed on one arm compared to the other. In this case, it is more informative to examine the boundaries on the partial sum scale, because that scale is interpretable for a two sample binomial proportions model as the difference in the number of events (treatment arm minus comparison arm).

1. Select the **Results** tab.
2. Set the **Display Scale** to **Partial Sum**.

From the command line, the same model can be selected and designed by typing:

```
> ssr.obf <- seqDesign(prob.model = "proportions",
+   arms = 2, size = 0.05, power = 0.95,
+   sample.size = 200, null.hypothesis = c(0.5, 0.5),
+   alt.hypothesis="calculate", variance = "null",
+   test.type = "two.sided", nbr.analyses = 4,
+   P = 1, display.scale = "S")
```

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.5, 0.5), alt.hypothesis = "calculate", variance
= "null", nbr.analyses = 4, sample.size = 200,
test.type = "two.sided", size = 0.05, power = 0.95, P =
1, display.scale = "S")

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0           (size = 0.05)
Alternative hypothesis : Theta > 0.2576 (power = 0.95)
(O'Brien & Fleming (1979))

STOPPING BOUNDARIES: Cumulative Sum scale
          a      b      c      d
Time 1 (N= 50) -14.3139     NA     NA 14.3139
Time 2 (N= 100) -14.3139     NA     NA 14.3139
Time 3 (N= 150) -14.3139     NA     NA 14.3139
Time 4 (N= 200) -14.3139 -14.3139 14.3139 14.3139
```

From the command line, you can display the same information about `ssr.obf` using the `print` function:

```
> print(ssr.obf)
```

This design has a power of .95 to detect a difference in toxicity rates of .2576. The trial would stop at the first analysis with a decision of harm by SSR only if there are 15 more deaths/toxicities on SSR than on placebo (out of 25 per arm).

The clinical researchers did not believe they would actually feel comfortable continuing with an unproven treatment in the face of such excess observed risk. Therefore, they considered a two-sided test with a less conservative lower boundary.

1. Select the **Advanced** tab and set **P** to `1.0, Inf, Inf, 0.5`. These values specify a Pocock boundary relationship on the upper (harm) boundary ($P = 0.5$) and an O'Brien-Fleming boundary relationship on the lower (efficacy) boundary ($P = 1.0$).
2. Select the Design tab and save the design object under the name `ssr.poc` using the **Save As** field.
3. Click **Apply**.

From the command line, the same model can be selected and designed by typing:

```
> ssr.poc <- update(ssr.obf, P=c(1.0,Inf,Inf,0.5))
```

Print results by typing:

```
> print(ssr.poc)
```

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis  
= c(0.5, 0.5), alt.hypothesis = "calculate", variance  
= "null", nbr.analyses = 4, sample.size = 200,  
test.type = "two.sided", size = 0.05, power = 0.95, P =  
c(1, Inf, Inf, 0.5), display.scale = "S")  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
Two-sided hypothesis test:  
Null hypothesis : Theta = 0 (size = 0.05)  
Alternative hypothesis : Theta > 0.2757 (power = 0.95)  
  
STOPPING BOUNDARIES: Cumulative Sum scale  
a b c d  
Time 1 (N= 50) -14.3139 NA NA 8.3485  
Time 2 (N= 100) -14.3139 NA NA 11.8065  
Time 3 (N= 150) -14.3139 NA NA 14.4600  
Time 4 (N= 200) -14.3139 -14.3139 16.6969 16.6969
```

This design has a power of .95 to detect a toxicity rate in the SSR arm which is .2757 higher than control arm. The Pocock boundary is less conservative at early analyses than the O'Brien–Fleming boundary—it stops at first analysis with harm by SSR given only 9 more toxicities on the SSR arm than on placebo.

This rate, however, was still considered excessive by the researchers, who subjectively wanted to stop with as few as 5 toxicities. This led to a third design which raised the upper boundary even further.

1. Select the **Advanced** tab and set **P** to **1.0,Inf,Inf,0.1**.
2. Select the **Design** tab and save the design object under the name `ssr.p0.1` using the **Save As** field.
3. Click **Apply**.

From the command line, the same model can be selected and designed by typing:

```
> ssr.p0.1 <- update(ssr.obf, P=c(1.0,Inf,Inf, 0.1))
```

Print results by typing:

```
> print(ssr.p0.1)
```

Unfortunately, the number of excess toxicities required to stop at the first analysis is still greater than 5.

STOPPING BOUNDARIES: Cumulative Sum scale				
	a	b	c	d
Time 1 (N= 50)	-14.3136	NA	NA	7.0710
Time 2 (N= 100)	-14.3136	NA	NA	13.1949
Time 3 (N= 150)	-14.3136	NA	NA	19.0060
Time 4 (N= 200)	-14.3136	-14.3136	24.6226	24.6226

Evaluating the Designs Let's plot the designs so far.

1. Select the **Plot** tab and choose the **Decision Boundaries** plot type.
2. In the **Designs To Plot** group box, click the **Refresh** button, then select `ssr.obf` and `SSR.POC` to compare with the current design (`ssr.p0.1`).
3. In the **Options** groupbox, set the **Display Scale** to **Partial Sum**.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(ssr.p0.1, ssr.obf, ssr.poc,
+     display.scale="S")
```

The result is shown in Figure 9.2.

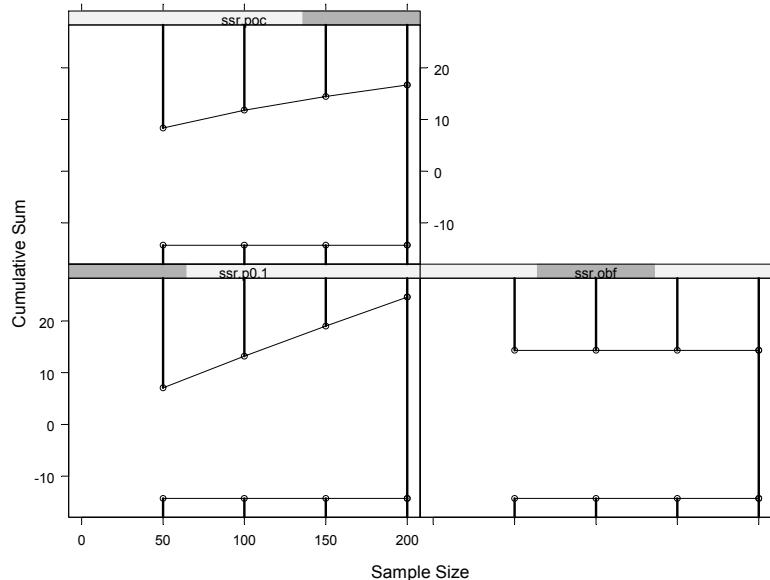


Figure 9.2: A comparison of stopping rules for two-sided tests.

The designs `ssr.obf`, `ssr.poc`, and `ssr.p0.1` all have approximately the same lower boundary, based on an O'Brien and Fleming relationship (note that changing the upper boundary in these two-sided sequential tests has only minimal effect on the lower boundary). They differ in the upper boundary, which is based on $P = 1$, 0.5 , and 0.1 , respectively.

Now compare the sample size distributions for these designs.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Average Sample Number** plot type.
3. In the **Options** groupbox, choose the **Overlay Designs** option.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotASN(ssr.p0.1, ssr.obf, ssr.poc, fixed=F)
```

The result is shown in Figure 9.3.

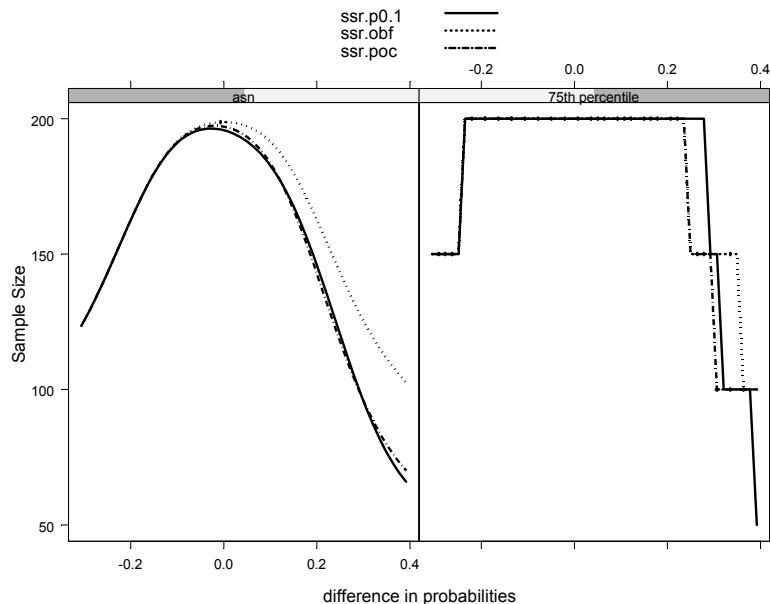


Figure 9.3: The average sample size (ASN) curve and the 75th %-tile curve for the two-sided tests $ssr.obf$, $ssr.poc$, and $ssr.p0.1$.

The $ssr.obf$ design has a higher expected sample size than the designs $ssr.poc$ and $ssr.p0.1$

Now compare the power curves.

1. Deselect the **Average Sample Number** plot type.
2. Select the **Power Curve** plot type.
3. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotPower(ssr.p0.1, ssr.obf, ssr.poc)
```

The result is shown in Figure 9.4.

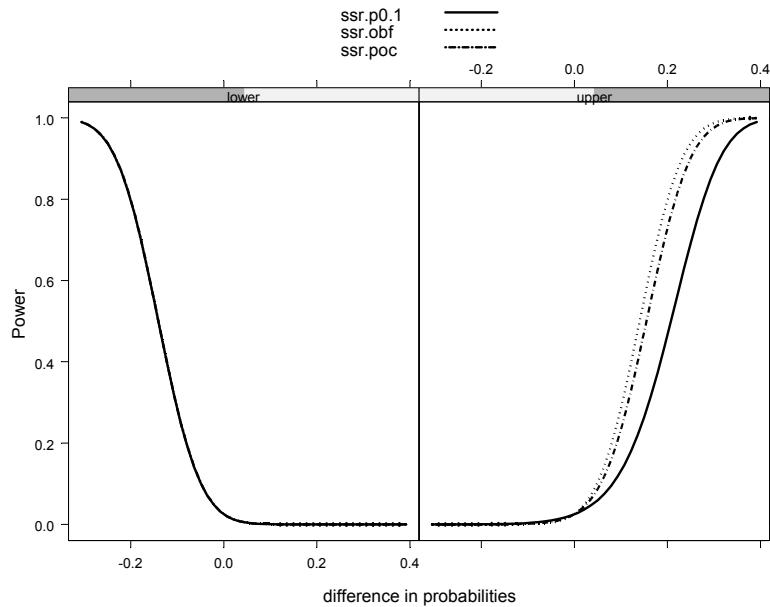


Figure 9.4: The power curves for the two sided tests `ssr.obf`, `ssr.poc`, and `ssr.p0.1`.

The design `ssr.p0.1` gives up a great deal of power to detect the upper hypothesis of harm by SSR even though the ASN curve is nearly identical to that of `ssr.poc`. You can see why this is the case by examining the stopping probabilities for the three designs.

1. Deselect the **Power Curve** plot type.
2. Select the **Stopping Probability** plot type.
3. Deselect the **Overlay Designs** option.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotStopProb(ssr.p0.1, ssr.obf, ssr.poc)
```

The result is displayed in Figure 9.5.

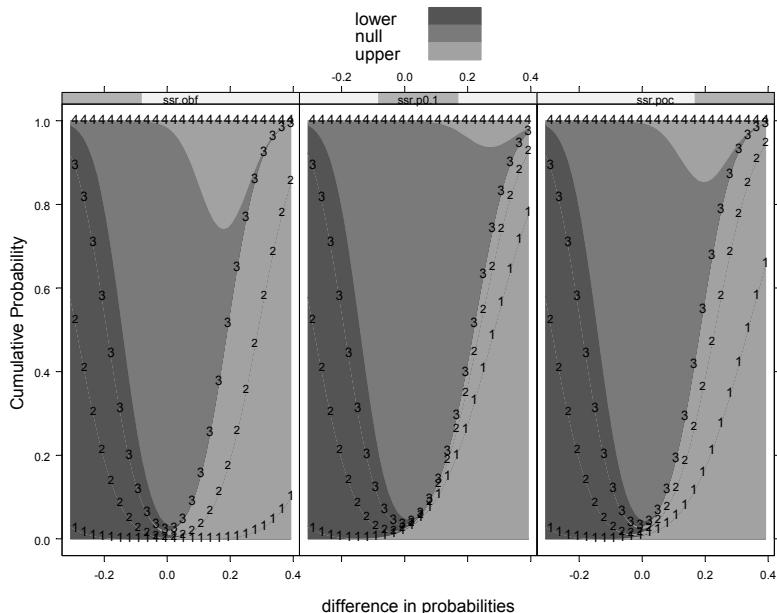


Figure 9.5: The stopping probabilities for the two sided tests `ssr.obf`, `ssr.p0.1`, and `ssr.poc`.

The designs `ssr.poc` and `ssr.p0.1` have a greater chance of stopping at the earliest two analyses than `ssr.obf`. Relative to `ssr.poc`, `ssr.p0.1` has very little chance of making a decision for SSR harm at the final analysis, even when the true treatment difference is quite positive.

In summary, the design with O'Brien-Fleming upper boundaries (`ssr.obf`) is distinctly different from the other two designs (`ssr.poc` and `ssr.p0.1`). Design `ssr.obf` has a higher expected sample size, higher power to detect harm by SSR, and a lower probability of stopping early in the trial. On the other hand, designs `ssr.poc` and `ssr.p0.1` have very similar sample size distributions, but design `ssr.p0.1` gives up a great deal of power to detect the upper alternative hypothesis of harm by SSR.

These designs demonstrate the limits to what can be achieved by adjusting the boundary shape parameter P . In the next section, we'll overcome these limitations by adjusting the epsilon shift parameter.

HYBRID TESTS

Although design `ssr.poc` is the best choice among the designs considered so far, this design still has two drawbacks. First, it requires 9 toxicities on SSR to stop the study at the first analysis, a difference the clinical researchers felt to be excessive. Second, the design has high power to decide that the SSR treatment is not sufficiently equivalent to the control treatment only if the difference between the true SSR toxicity rate and the control toxicity rate were 30%.

This turned out to be a problem shared by all the two-sided designs. A lower boundary relationship which is even less conservative at the first analysis ($P = .1$) *still* did not achieve the clinician's subjective desire for stopping if an excess of 5 deaths were observed on the SSR arm, and the power to detect harm due to SSR declined even further.

Hence, stopping rules intermediate to two-sided and one-sided designs were explored. With appropriate choices of the shift parameters, such a design can be thought of as a hybrid between the upper boundary on a one-sided equivalence (noninferiority) design and the lower (efficacy) boundary on a one-sided test of a lesser alternative. This design then satisfies the need to be able to decide superiority for SSR, as well as deciding lack of approximate equivalence between the two treatment arms. It does not suffer the drawbacks of having to prove that the new, unproven SSR is worse than the standard therapy.

Using the Epsilon Shift Parameter

The epsilon shift parameter allows you to define a range of hybrid tests that form a continuum between one-sided and two-sided tests. A vector of length two represents the upward shift of the lower hypothesis test ε_l and the downward shift of the upper hypothesis test ε_u . If both hypothesis shift parameters are equal to one (so the sum of those parameters is equal to two), the design corresponds to a two-sided test. If the sum of the hypothesis shift parameters equals one, the design corresponds to a one-sided test. A one-sided test tests against positive alternatives if $\varepsilon_l = 0$, and against negative alternatives if $\varepsilon_u = 0$. If $\varepsilon_l = \varepsilon_u = 0.5$, this corresponds to a test

suitable for establishing one-sided equivalence (noninferiority) between treatments results, providing the study is designed with sufficiently high power. (For more information, see page 110.)

To achieve the design criteria in this case study, try shifting hypotheses using the epsilon parameter.

1. Close the current S+SEQTRIAL dialog.
2. Choose **SeqTrial ► Update Design**.
3. Select **ssr.poc** from the listbox and click **OK**.
4. Save the new design under the name **ssr.eps.8** using the **Save As** field.
5. Select the **Advanced** tab and set the **Epsilon** field to **1, 0.8**.
6. Click **Apply**.

From the command line, the same model can be selected and designed by typing:

```
> ssr.eps.8 <- update(ssr.poc, test.type="advanced",
+ epsilon=c(1, 0.8))
```

Print results by typing:

```
> print(ssr.eps.8)
```

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.5, 0.5), alt.hypothesis = "calculate", variance
= "null", nbr.analyses = 4, sample.size = 200,
test.type = "two.sided", size = 0.05, power = 0.95, epsilon =
c(1, 0.8), P = c(1, Inf, Inf, 0.5), display.scale = "S")
```

PROBABILITY MODEL and HYPOTHESES:

Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Hybrid hypothesis test:

Null hypothesis : Theta = 0
Alternative hypothesis : Theta > 0.2137 (power = 0.95)

STOPPING BOUNDARIES: Cumulative Sum scale

	a	b	c	d
Time 1 (N= 50)	-14.3135	NA	NA	6.7979
Time 2 (N= 100)	-14.3135	NA	NA	8.7055
Time 3 (N= 150)	-14.3135	NA	NA	9.8084
Time 4 (N= 200)	-14.3135	-14.3135	10.4949	10.4949

Setting the epsilon parameter to 1, 0.8 specifies a design intermediate between a two-sided test and a one-sided test for the upper alternative.

This design has a power of .95 to detect a difference in toxicities in excess of .2137, but still requires 7 more toxicities on SSR than on placebo to stop at the first analysis. Therefore, let's try moving closer to a one-sided design.

1. Set **Epsilon** to **1, 0.5**.
2. Select the **Design** tab. Note that the **Test Type** is now shown as **advanced** in the **Hypotheses** groupbox.
3. Save the new design under the name **ssr.eps.5** using the **Save As** field.
4. Click **Apply**.

From the command line, the same model can be selected and designed by typing:

```
> ssr.eps.5 <- update(ssr.poc, test.type="advanced",
+ epsilon=c(1, 0.5))
```

Print results by typing:

```
> print(ssr.eps.5)
```

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.5, 0.5), alt.hypothesis = "calculate", variance
= "null", nbr.analyses = 4, sample.size = 200,
test.type = "advanced", size = 0.05, power = 0.95,
epsilon = c(1, 0.5), P = c(1, Inf, Inf, 0.5),
display.scale = "S")
```

```
PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Hybrid hypothesis test:
Null hypothesis : Theta = 0
Alternative hypothesis : Theta > 0.1207      (power = 0.95)
```

Chapter 9 Case Study #1: Hybrid Tests

```
STOPPING BOUNDARIES: Cumulative Sum scale
                    a      b      c      d
Time 1 (N= 50) -14.307    NA    NA 4.4730
Time 2 (N= 100) -14.307    NA    NA 4.0555
Time 3 (N= 150) -14.307    NA    NA 2.8335
Time 4 (N= 200) -14.307 -14.307 1.195 1.1950
```

This design stops at the first analysis with 5 more toxicities on SSR than on placebo. Now try going even closer to a one-sided design.

1. Save the new design under the name ssr.eps.2 using the Save As field.
2. Select the **Advanced** tab and set the **Epsilon** field to **1, 0.2**.
3. Click **Apply**.

From the command line, the same model can be designed and printed by typing:

```
> ssr.eps.2 <- update(ssr.poc, test.type="advanced",
+   epsilon=c(1, 0.2))
> print(ssr.eps.2)
```

Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.5, 0.5), alt.hypothesis = "calculate", variance
= "null", nbr.analyses = 4, sample.size = 200,
test.type = "advanced", size = 0.05, power = 0.95,
epsilon = c(1, 0.2), P = c(1, Inf, Inf, 0.5),
display.scale = "S")

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Hybrid hypothesis test:
Null hypothesis : Theta = 0
Alternative hypothesis : Theta > 0.02846 (power = 0.95)

```
STOPPING BOUNDARIES: Cumulative Sum scale
                    a      b      c      d
Time 1 (N= 50) -14.2224    NA    NA 2.1640
Time 2 (N= 100) -14.2224    NA    NA -0.5619
Time 3 (N= 150) -14.2224    NA    NA -4.0922
Time 4 (N= 200) -14.2224 -14.2224 -8.0389 -8.0389
```

This design stops at first analysis with only 3 more toxicities on SSR than on placebo. It has a power of 0.95 to detect a difference in toxicities in excess of 0.02846.

Finally, for the sake of comparison, let's also compute a one-sided design.

1. Set **Epsilon** to 1, 0.
2. Select the **Design** tab and save the new design under the name **ssr.oneside** using the **Save As** field.
3. Click **Apply**.

From the command line, the same model can be designed and printed by typing:

```
> ssr.oneside <- update(ssr.poc, test.type="advanced",
+   epsilon=c(1, 0))
> print(ssr.oneside)
```

Evaluating the Designs

Plot the boundaries for all of the designs.

1. Select the **Plot** tab and choose the **Decision Boundaries** plot type.
2. In the **Designs to Plot** listbox, click the **Refresh** button, then select **ssr.obf**, **ssr.poc**, **ssr.eps.8**, **ssr.eps.5**, and **ssr.eps.2** to compare with the current design (**ssr.oneside**).
3. Set the **Display Scale** to **Partial Sum**.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> plot(ssr.oneside, ssr.obf, ssr.poc, ssr.eps.8,
+   ssr.eps.5, ssr.eps.2)
```

The result is shown in Figure 9.6.

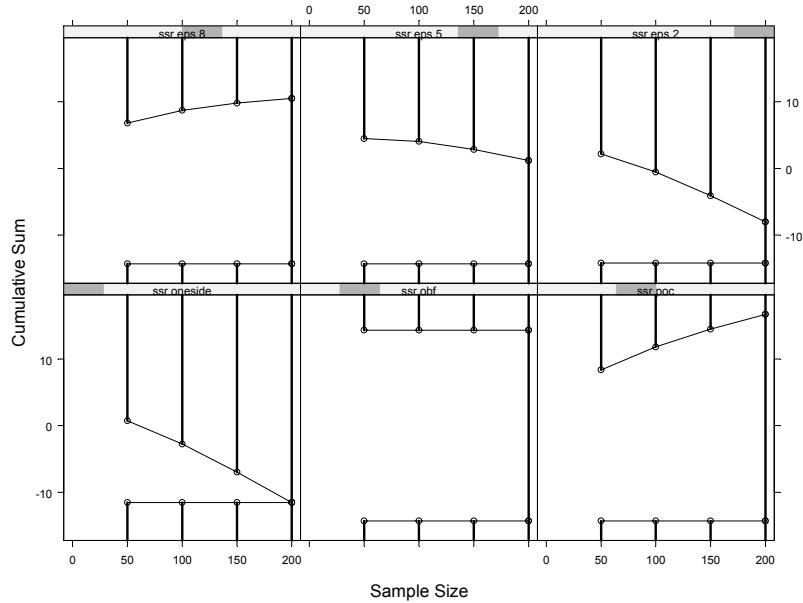


Figure 9.6: A comparison of stopping rules for two-sided tests, hybrid tests (in between one-sided and two-sided), and a one-sided test.

Recall that the designs `ssr.obf` and `ssr.poc` are two-sided tests, the design `ssr.oneside` is a one-sided test, and the designs `ssr.eps.8`, `ssr.eps.5` and `ssr.eps.2` are hybrid tests, intermediate between one-sided and two-sided tests.

Plot the ASN and power curves.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Average Sample Number** and **Power Curve** plot types.
3. Select the **Overlay Designs** option.
4. Click **Apply**.

From the command line, the same plots can be created by typing:

```
> seqPlotASN(ssr.oneside, ssr.obf, ssr.poc,
+   ssr.eps.8, ssr.eps.5, ssr.eps.2, fixed=F)

> seqPlotPower(ssr.oneside, ssr.obf, ssr.poc,
+   ssr.eps.8, ssr.eps.5, ssr.eps.2)
```

The results are shown in Figure 9.7 and Figure 9.8.

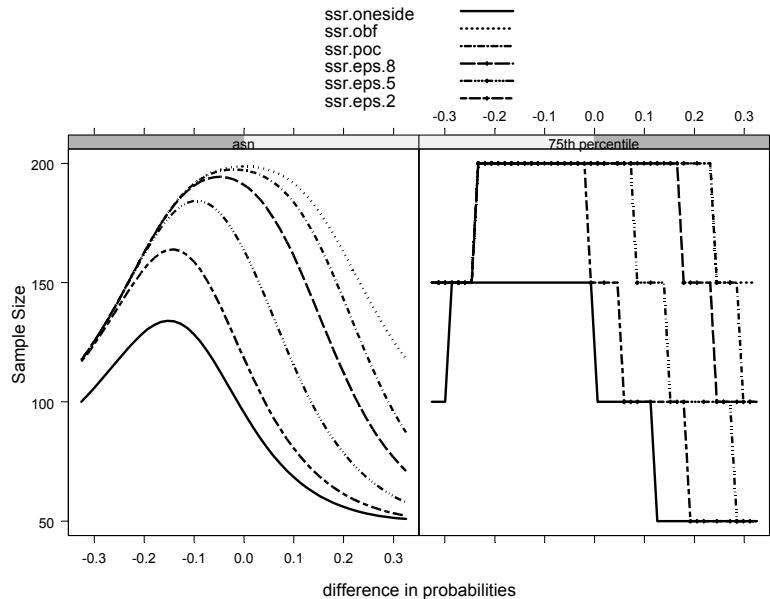


Figure 9.7: The average sample size (ASN) curve and the 75th %-tile curve for the two sided tests `ssr.obf` and `ssr.poc`, the one-sided test `ssr.oneside`, and the hybrid tests `ssr.eps.8`, `ssr.eps.5` and `ssr.eps.2`.

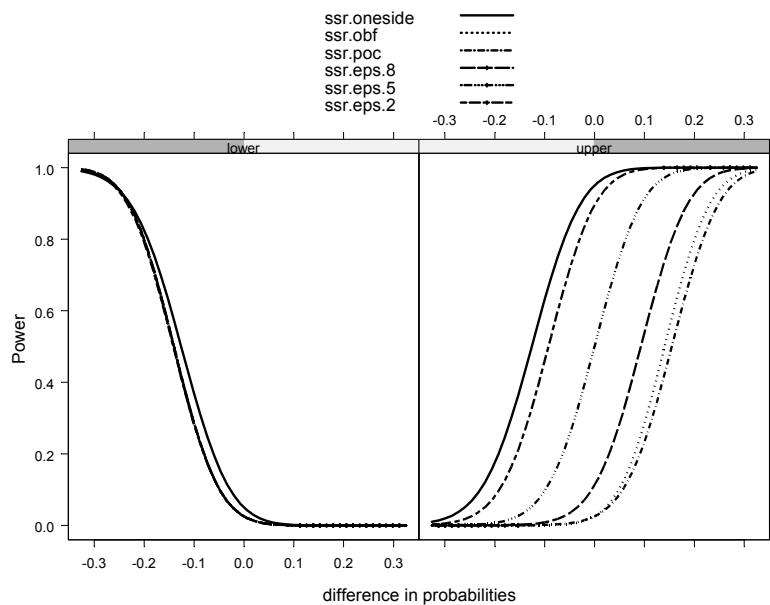


Figure 9.8: The power curves for the two sided tests `ssr.ofb` and `ssr.poc`, the one-sided test `ssr.oneside`, and the hybrid tests `ssr.eps.8`, `ssr.eps.5` and `ssr.eps.2`.

The design `ssr.eps.5` seems to best meet the researcher's criteria.

Hybrid tests allow a continuum of design choices between two-sided and one-sided tests. In this case study, it was possible to achieve the researchers design criteria by adjusting the epsilon shift parameter. The design criteria were not attainable by simply adjusting the boundary shape rules for two-sided (or one-sided) tests.

CASE STUDY #2: UNPLANNED ANALYSES

10

Overview	262
Designing a Fixed Sample Test	263
Selecting the Model	263
Specifying the Design	263
Designing a Group Sequential Test	267
Specifying the Design	267
Evaluating the Design	268
Conducting the Clinical Trial	272
Recreating Data for the First Analysis	272
Monitoring at the First Analysis	273
Evaluating the Modified Design	274
Recreating Data for the Second Analysis	278
Monitoring at the Second Analysis	279
Evaluating the Modified Design	281
Performing a Sensitivity Analysis	283
Boundary Shift Function	283
Computing the Best and Worst Case Designs	284
Evaluating the Designs	286

OVERVIEW

Emerson & Banks (1994) describe the case of a clinical trial conducted to compare a new drug (idarubicin) to an existing drug (daunorubicin) in the treatment of acute myelogenous leukemia. Although no formal monitoring rule was pre-specified, during the conduct of the study the researchers observed promising trends in favor of idarubicin over daunorubicin in inducing complete remission. Partway into the study, a formal stopping rule was retrospectively imposed. Eventually, the trial was stopped early with a decision in favor of the new treatment, not only with respect to the primary outcome of induction of remission, but also with respect to overall survival. However, when the pharmaceutical company sought FDA approval for the marketing of idarubicin, concerns were raised about the possibility that the retrospective imposition of the stopping rule might invalidate the statistical inference.

The problem that needs to be addressed is how to draw statistical inference in such a setting. One approach to this problem is to perform a sensitivity analysis which examines a range of stopping rules which might reasonably have been the true sampling density for the estimate of treatment effect. By examining the stability of the statistical inference over these varied stopping rules, some insight is gained into whether we can have high confidence in the scientific conclusions reached by this clinical trial. This case study illustrates the advantages that a flexible computing environment like S+SEQTRIAL provides when conducting such a specialized analysis of clinical trial results.

Note that the data analyses in this case study have been slightly modified from those reported by Emerson & Banks (1994). At the time of the original clinical trial, the methods available in S+SEQTRIAL were not fully developed, and more approximate techniques were used.

DESIGNING A FIXED SAMPLE TEST

The original study protocol called for a single formal analysis of the data following the accrual of the entire planned sample size.

Selecting the Model

In this study, a binary endpoint measured benefit as induction of a complete remission of the leukemia following treatment, with subjects assigned with equal probability to the idarubicin treatment group and to the daunorubicin treatment group. Hence, the model is a two-sample Binomial hypothesis test. Choose **SeqTrial ► Two Samples ► Binomial Proportions** to launch the appropriate S+SEQTRIAL dialog for this trial.

All of the commands needed to run this case study from the command line are contained in a help file. Type:

```
> help(case2.seqtrial)
```

Specifying the Design

Under the null hypothesis, the remission rates were assumed to be the same. The baseline remission rate was taken to be 60%. The sample size was calculated so that the test would have power of 0.8 to detect differences in remission rates $\geq .2$.

Let's specify this design.

1. Ensure that the **Sample Size** option is selected for the computational task (the default).
2. Set the **Significance Level** is to 0.05 and the **Power** to 0.80.
3. The remission rate under the null hypothesis is 0.60, so set the **Null Proportions** field to 0.60, 0.60 (or just 0.60). Set the **Alt Proportions** field to 0.80, 0.60 (or just 0.80).
4. Ensure that the **Variance Method** is set to alternative.
5. Set the **Test Type** to **two.sided**.
6. Save the design object under the name **aml.fix** using the **Save As** field.

The **Design** tab should now look like that shown in Figure 10.1.

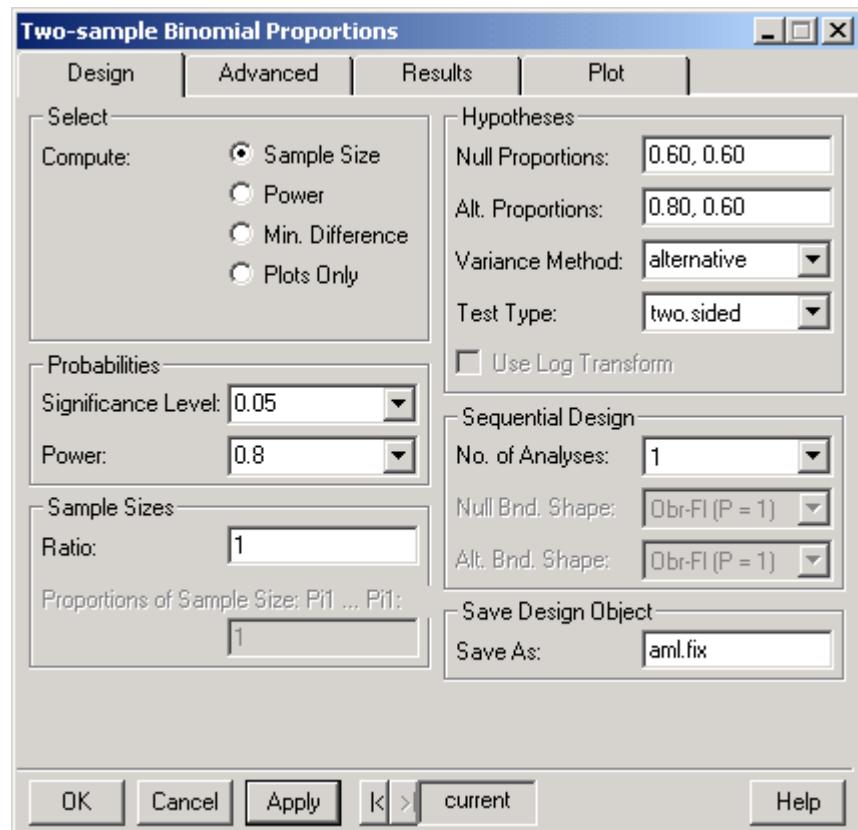


Figure 10.1: *Design* tab for a fixed sample test.

From the command line, the same model can be selected and designed by typing:

```
> aml.fix <- seqDesign(prob.model = "proportions",
+   arms = 2, size = 0.05, power = 0.80,
+   null.hypothesis = c(0.60, 0.60),
+   alt.hypothesis = c(0.80, 0.60),
+   variance = "alternative",
+   test.type = "two.sided")
```

Click **Apply** to create the design object and print out summary information in a report window.

```

Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.6, 0.6), alt.hypothesis = c(0.8, 0.6), ratio =
c(1., 1.), nbr.analyses = 1, test.type = "two.sided",
power = 0.8, alpha = 0.025, beta = 0.975, epsilon = c(
1., 1.), early.stopping = "both", display.scale =
seqScale(scaleType = "X"))

PROBABILITY MODEL and HYPOTHESES:
  Two arm study of binary response variable
  Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
  Null hypothesis : Theta = 0      (size = 0.05 )
  Alternative hypothesis : Theta > 0.2   (power = 0.8 )
  (Fixed sample test)

STOPPING BOUNDARIES: Sample Mean scale
          a      b      c      d
Time 1 (N= 156.98) -0.1399 -0.1399 0.1399 0.1399

```

From the command line, you can display the same information about `aml.fix` using the `print` function:

```
> print(aml.fix)
```

This design requires 157 subjects. The sponsor wished to revise this upward to 180 patients in order to ensure sufficient evaluable patients for various secondary analyses.

1. Select the **Min. Difference** option for the computational task.
2. Set the **N** field in the **Sample Sizes** groupbox to 180.

From the command line, the same model can be selected and designed by typing:

```
> aml.fix <- seqDesign(prob.model = "proportions",
+   arms = 2, size = 0.05, power = 0.80,
+   null.hypothesis = c(0.60, 0.60),
+   alt.hypothesis = "calculate",
+   variance = "alternative",
+   test.type = "two.sided", sample.size=180)
```

Click **Apply** to create the design object and print out summary information in a report window.

Chapter 10 Case Study #2: Unplanned Analyses

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis  
= c(0.6, 0.6), alt.hypothesis = "calculate", ratio =  
c(1., 1.), nbr.analyses = 1, sample.size = 180,  
test.type = "two.sided", power = 0.8, alpha = 0.025,  
beta = 0.975, epsilon = c(1., 1.), early.stopping =  
"both", display.scale = seqScale(scaleType = "X"))  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
Two-sided hypothesis test:  
Null hypothesis : Theta = 0 (size = 0.05 )  
Alternative hypothesis : Theta > 0.1884 (power = 0.8 )  
(Fixed sample test)  
  
STOPPING BOUNDARIES: Sample Mean scale  
a b c d  
Time 1 (N= 180) -0.1318 -0.1318 0.1318 0.1318
```

From the command line, you can display the same information about `aml.fix` using the `print` function:

```
> print(aml.fix)
```

Because of the additional subjects, this design has a power of .8 to detect differences in remission rates $\geq .1884$.

DESIGNING A GROUP SEQUENTIAL TEST

After the trial had started, early informal and unplanned analyses of the data suggested some advantage of idarubicin over daunorubicin in inducing complete remission. Therefore, after accrual of a total of 69 patients, a level 0.05 O'Brien–Fleming group sequential design with a total of three formal analyses was retroactively adopted for the study. The interim analyses were scheduled after accrual of 90 and 130 patients (45 and 65 per treatment arm).

Specifying the Design This design can be specified as follows:

1. Set the **Number of Analyses** to 3 on the **Design** tab. This specifies the total number of analyses (interim plus final).
2. Set the **Null Boundary Shape** to **No Early Stopping**.
3. Save the design object under the name **aml.obf** using the **Save As** field.
4. Set the **N (or N1 ... N3)** field to **90, 130, 180**.

From the command line, the same model can be selected and designed by typing:

```
> aml.obf <- update(aml.fix, nbr.analyses=3,
+   early.stopping="alternative",
+   sample.size=c(90,130,180))
```

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.6, 0.6), alt.hypothesis = "calculate", ratio =
c(1., 1.), nbr.analyses = 3, sample.size = c(90, 130,
180), test.type = "two.sided", power = 0.8, alpha =
0.025, beta = 0.975, epsilon = c(1., 1.), display.scale
= seqScale(scaleType = "X"))

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0          (size = 0.05 )
Alternative hypothesis : Theta > 0.1903    (power = 0.8 )
```

Chapter 10 Case Study #2: Unplanned Analyses

(O'Brien & Fleming (1979))

```
STOPPING BOUNDARIES: Sample Mean scale
      a      b      c      d
Time 1 (N= 90) -0.2713     NA     NA 0.2713
Time 2 (N= 130) -0.1878     NA     NA 0.1878
Time 3 (N= 180) -0.1356 -0.1356 0.1356 0.1356
```

From the command line, display the same information by typing:

```
> print(aml.obf)
```

Note that the addition of interim analyses has only slightly increased the minimum detectable difference in remission rates (0.1903 versus 0.1884).

Evaluating the Design

Let's plot the design.

1. Select the **Plot** tab and choose the **Decision Boundaries** plot type.
2. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(aml.obf, fixed=F)
```

The result is shown in Figure 10.2.

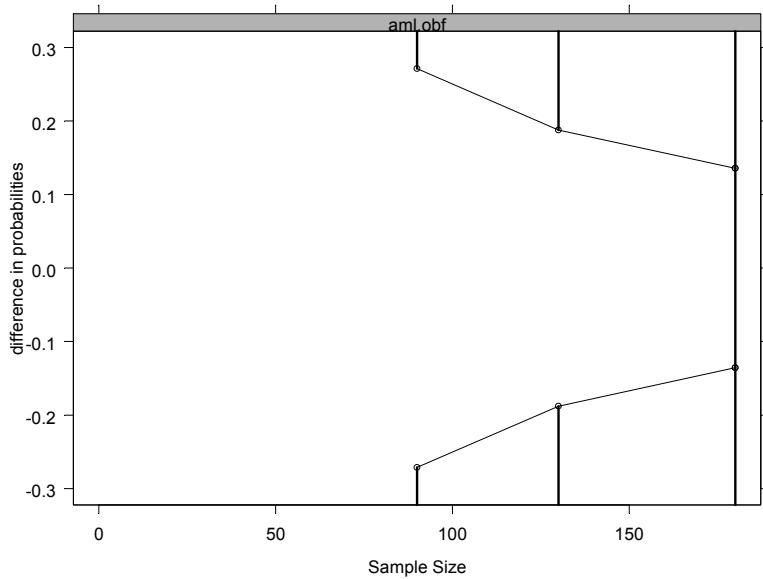


Figure 10.2: The stopping rules for an O'Brien–Fleming group sequential design.

Now let's examine the change in power due to adding interim analyses.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Power Curve** plot.
3. In the **Designs to Plot** listbox, click the **Refresh** button, then select the fixed design object `aml.fix`.
4. Select **Overlay Designs** in the **Options** groupbox.

Click **Apply**. Figure 10.3 shows the result. The power curves are visually indistinguishable.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(aml.obf, aml.fix, superpose.design=T)
```

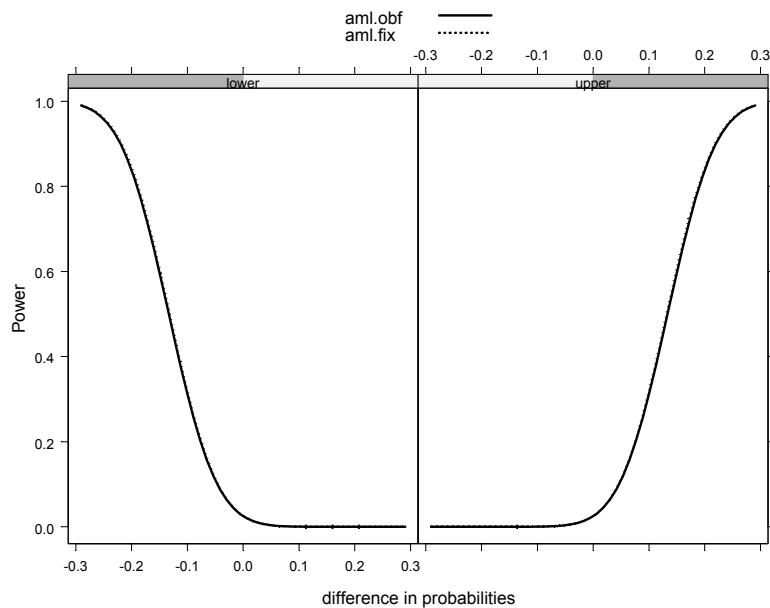


Figure 10.3: The power curves for the group sequential design are visually indistinguishable from the power curves for the equivalent fixed sample test.

It is easier to compare power curves by plotting the difference between them.

1. Deselect the **Overlay Designs** and **Power Curve** options.
2. Select the **Power Curve vs. Reference** plot type.
3. Set the **Reference Design** to `aml.fix`.

Click **Apply**. The result in this case is shown in Figure 10.4.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(aml.obf, aml.fix, reference=aml.fix)
```

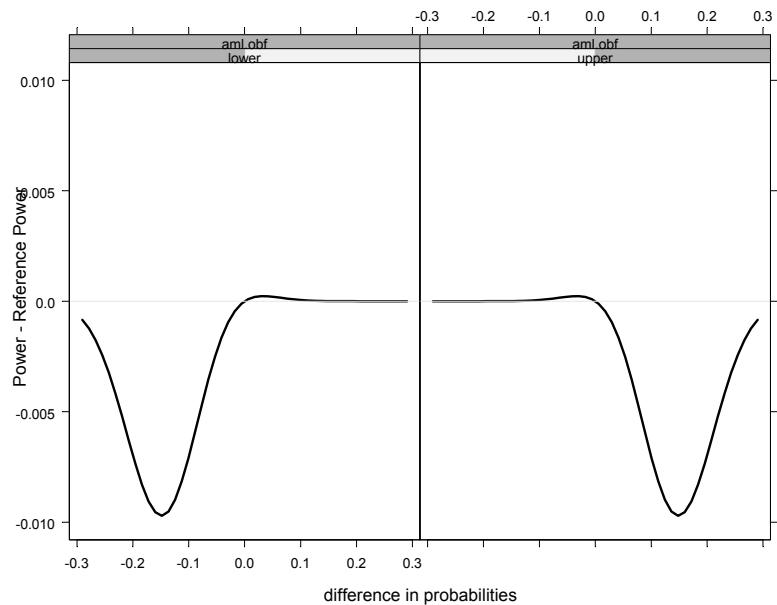


Figure 10.4: The difference curve created by subtracting the fixed design from the group sequential design.

You can see that the maximum difference in the two power curves is approximately -.01, and usually far less.

CONDUCTING THE CLINICAL TRIAL

The first formal interim analysis took place after 90 patients had been accrued to the clinical trial (45 patients to each arm). 35 out of 45 patients (78%) in the idarubicin arm had achieved a complete remission compared to 25 out of 45 (56%) on the daunorubicin arm.

To add this information to our design, we can make use of the trial monitoring capabilities of S+SEQTRIAL.

The seqMonitor function computes modified boundaries so that the Type I statistical error is maintained at the level specified during the design stage. Additionally, you can decide whether to maintain the maximal sample size or the power of the study.

The seqMonitor function takes a design, a vector of response measurements for the patients in the trial, and a vector indicating treatment assignment for each of the corresponding patients. In a real setting, the data regarding patient response and treatment assignment would be extracted from the clinical trial data base, but for the purposes of this case study, we'll just recreate the data.

For more information on monitoring, see Chapter 7.

Recreating Data for the First Analysis

For the idarubicin arm, let's create a vector having 35 values of 1, to reflect the 35 patients with complete remission, and 10 values of 0, to reflect the remaining 10 patients on that arm who did not have a complete remission. Select the **Commands** window and enter

```
> cr.idr <- c(rep(1,35), rep(0,10))
```

For the daunorubicin arm, create a vector having 25 values of 1, and 20 values of 0.

```
> cr.dnr <- c(rep(1,25), rep(0,20))
```

Combine these results into a vector reflecting the total sample.

```
> cr1 <- c(cr.idr, cr.dnr)
```

Finally, create a vector of treatment indicators in which 1 denotes a patient on the idarubicin arm, and 0 a patient on the daunorubicin arm.

```
> tx1 <- c(rep(1,45), rep(0, 45))
```

Monitoring at the First Analysis

The seqMonitor function computes modified stopping boundaries based on revised estimates of response measurement variability, taking into account whether the number and timing of analyses that occur during the actual conduct of a group sequential trial vary from the number and timing of analyses estimated during the design phase. In this case, the timing of the first analysis was precisely as planned, but the stopping rule must be modified to reflect the better estimates of the remission rates on each arm that are available at the interim analysis.

You can specify the operating characteristics that the seqMonitor function maintains when revising the boundaries. Type I error is always maintained, but the statistical power to detect the design alternative or the maximal sample size can also be preserved. Let's first try maintaining the statistical power for the design alternative, since this is the default. Typically, we can expect some modification of the sample size as a result of errors in estimating the response rates at the design stage.

To monitor this trial, enter the **Sequential Monitor** dialog,

1. Type **aml.obf** in the **Previous Design** field.
2. Type **cr1** in the **Response Vector** field.
3. Type **tx1** in the **Treatment Vector** field.
4. Type **first.analysis** in the **Save As** field.

Click **Apply**. The result is:

```

Call:
seqMonitor(x = aml.obf, response = cr1, treatment = tx1,
constraint.scale = "X", min.increment = 0.1, maxiter = 20)

RECOMMENDATION:
Continue

OBSERVED STATISTICS:
Sample Size Crude Estimate Z Statistic
         90          0.2222        2.301

MONITORING BOUNDS:
Call:
"(not shown, is (your seqMonitor object)$seqDesignCall)"

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable

```

Chapter 10 Case Study #2: Unplanned Analyses

```
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
  Null hypothesis : Theta = 0           (size  = 0.05)
  Alternative hypothesis : Theta > 0.1898   (power = 0.8 )
(O'Brien & Fleming (1979))

STOPPING BOUNDARIES: Sample Mean scale
      a      b      c      d
Time 1 (N= 90) -0.2812    NA    NA 0.2812
Time 2 (N= 136) -0.1861    NA    NA 0.1861
Time 3 (N= 187) -0.1353 -0.1353 0.1353 0.1353
```

At this first analysis, the recommendation is to continue trial. Early stopping is not appropriate.

From the command line, you can obtain the same results using:

```
> first.analysis <- seqMonitor(am1.obf, cr1, tx1)
> print(first.analysis)
```

Evaluating the Modified Design

Let's plot the modified stopping rule and the current estimate of treatment effect. On the **Plot** tab of the **Sequential Monitor** dialog,

1. Select **Decision Boundaries**.
2. Select **Overlay Designs**.

Click **Apply**. The result is shown in Figure 10.5. Note that the observed data is not outside the upper boundary.

From the command line, type:

```
> plot(first.analysis)
```

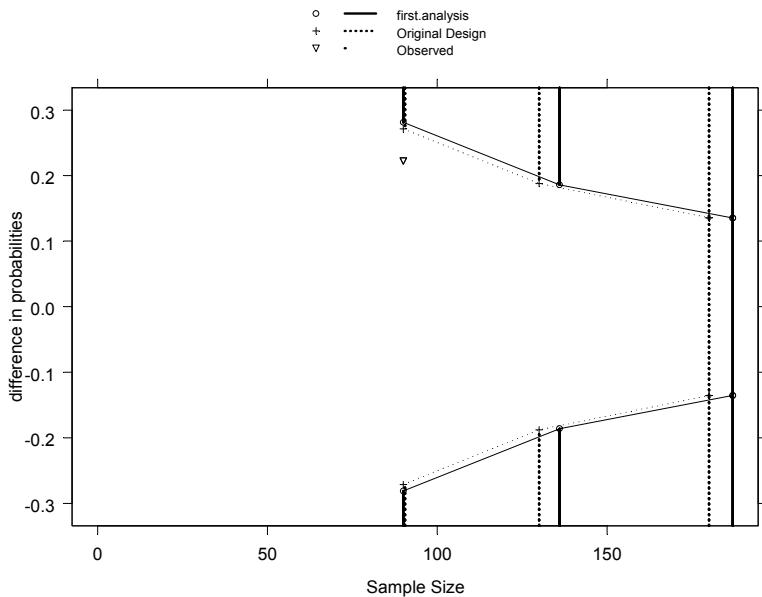


Figure 10.5: The modified stopping boundaries and the current estimate of treatment effect at the first interim analysis.

The original stopping rule estimated at the design stage is also plotted in Figure 10.5. A larger sample size is now required to maintain 80% power to detect a treatment effect of .1903. This is primarily due to the fact that the response rate on the daunorubicin arm was observed to be 55.6%, less than the 60% estimated at the design stage, resulting in more variable measurement of response. (Binomial data is more variable the closer the event rate is to .5).

The sponsor did not want to vary from the planned maximal sample size of 180 subjects, so the monitoring plan actually used reflected the desire to maintain that sample size, perhaps at the cost of some statistical power. To produce this monitoring design

On the **Design** tab of the **Sequential Monitor** dialog,

1. Type 130, 180 in the **Future Analyses** field.

Chapter 10 Case Study #2: Unplanned Analyses

Click **Apply**.

From the command line, type:

```
> first.analysis <- seqMonitor(aml.obf, crl, tx1,
+     future.analyses=c(130,180))
> print(first.analysis)
```

Call:

```
seqMonitor(x = aml.obf, response = crl, treatment = tx1,
future.analyses = c(130, 180), constraint.scale = "X",
min.increment = 0.1, maxiter = 20)
```

RECOMMENDATION:

Continue

OBSERVED STATISTICS:

Sample	Size	Crude Estimate	Z Statistic
90		0.2222	2.301

MONITORING BOUNDS:

Call:

```
"(not shown, is (your seqMonitor object)$seqDesignCall)"
```

PROBABILITY MODEL and HYPOTHESES:

Two arm study of binary response variable

Theta is difference in probabilities (Treatment - Comparison)

Two-sided hypothesis test:

Null hypothesis : Theta = 0 (size = 0.05)

Alternative hypothesis : Theta > 0.1935 (power = 0.8)

(O'Brien & Fleming (1979))

STOPPING BOUNDARIES: Sample Mean scale

	a	b	c	d
Time 1 (N= 90)	-0.2759	NA	NA	0.2759
Time 2 (N= 130)	-0.1910	NA	NA	0.1910
Time 3 (N= 180)	-0.1380	-0.138	0.138	0.1380

Again, the recommendation is to continue. Note the slight loss in power. We now have power of .8 to detect a difference of 0.1935, instead of 0.1903.

Assuming that you still had **Decision Boundaries** and **Overlay Designs** selected on the plot menu, the plot shown in Figure 10.6 should appear.

From the command line, type:

```
> plot(first.analysis)
```

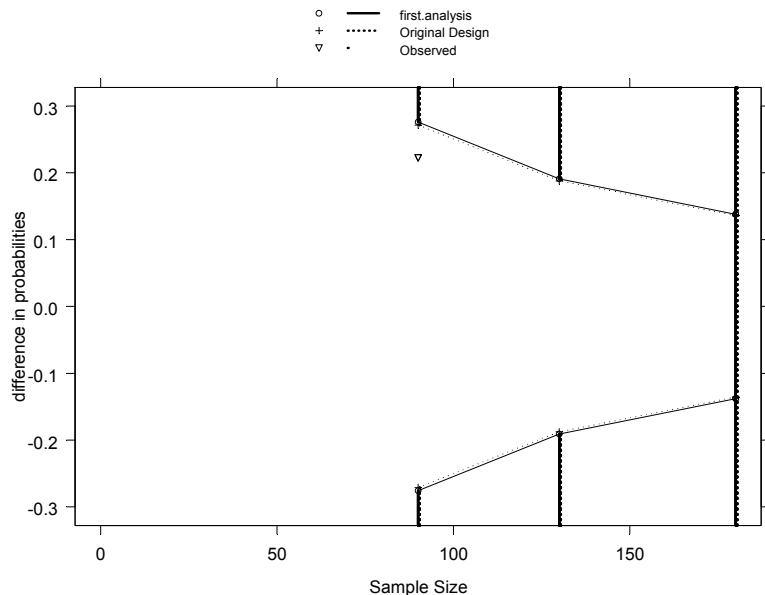


Figure 10.6: A comparison of the modified stopping rule to what was estimated at the design stage.

The modified and original boundaries are virtually identical.

Let's examine the loss of power graphically. Because seqMonitor objects inherit from seqDesign objects, they can be passed to the seqPlotPower function.

On the **Plot** tab of the **Sequential Monitor** dialog,

1. Deselect **Decision Boundaries**.
2. Select **Power Curve vs. Reference**.
3. Type **aml.obf** in the **Reference Design** field.

Click **Apply**.

From the command line, type:

```
> seqPlotPower(first.analysis, aml.obf,
+               reference="aml.obf", superpose.design=T)
```

The result is shown in Figure 10.7.

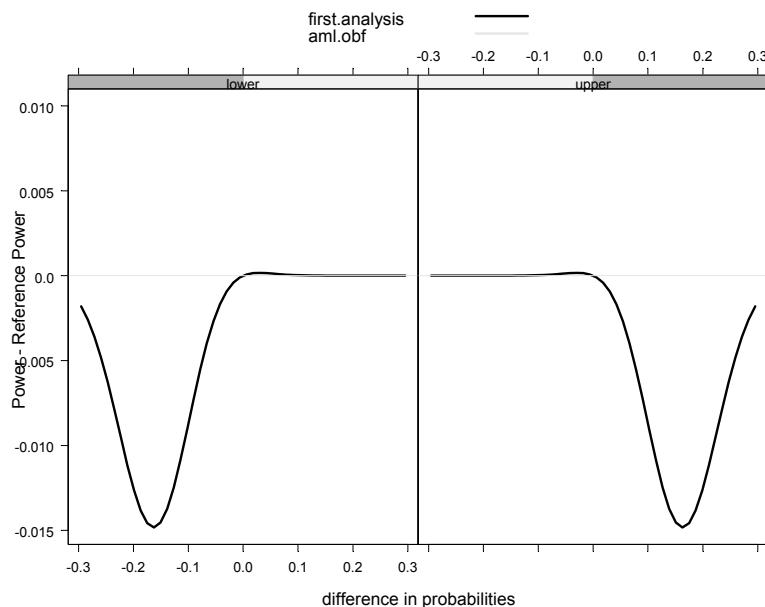


Figure 10.7: The small loss of power due to the modified stopping boundaries.

The maximum loss of power is only about -0.015, and it is usually less.

Recreating Data for the Second Analysis

The decision made at the first analysis was that the trial should be continued. An additional 40 patients were accrued to the study (20 on each of the arms). Of the 20 new patients accrued to the idarubicin arm, 16 achieved a complete remission. 13 of the 20 new patients accrued to the daunorubicin arm achieved a complete remission. With a total of 130 patients' data available for analysis, it was time to perform the second formal analysis.

Again, in a real setting the data regarding patient response and treatment assignment would be extracted from the clinical trial data base. For the purposes of this case study, however, we must recreate that data.

For the idarubicin arm, create a vector having 16 values of 1, to reflect the 16 new patients with complete remission, and 4 values of 0, to reflect the remaining 4 new patients on that arm who did not have a complete remission.

```
> cr.idr <- c(rep(1,16), rep(0,4))
```

For the daunorubicin arm, create a vector having 13 values of 1, and 7 values of 0.

```
> cr.dnr <- c(rep(1,13), rep(0,7))
```

Combine these results and the results from the first analysis into a vector reflecting the current trial experience.

```
> cr2 <- c(cr1, cr.idr, cr.dnr)
```

Finally, create a vector of treatment indicators in which 1 denotes a patient on the idarubicin arm, and 0 a patient on the daunorubicin arm.

```
> tx2 <- c(tx1, rep(1,20), rep(0, 20))
```

Monitoring at the Second Analysis

To conduct the second interim analysis, on the **Design** tab of the **Sequential Monitor** dialog:

1. Type **first.analysis** in the **Previous Design** field.
2. Type **cr2** in the **Response Vector** field.
3. Type **tx2** in the **Treatment Vector** field.
4. Type **180** in the **Future Analyses** field.
5. Type **second.analysis** in the **Save As** field.

Click **Apply**.

From the command line, type:

```
> second.analysis <- seqMonitor(first.analysis, cr2,
+      tx2, future.analyses=180)
> print(second.analysis)
```

Chapter 10 Case Study #2: Unplanned Analyses

```
Call:  
seqMonitor(x = first.analysis, response = cr2, treatment = tx2,  
future.analyses = 180, constraint.scale = "X",  
min.increment = 0.1, maxiter = 20)  
  
RECOMMENDATION:  
Stop with decision for Upper Alternative Hypothesis  
  
OBSERVED STATISTICS:  
Sample Size Crude Estimate Z Statistic  
90 0.2222 2.301  
130 0.2000 2.513  
INFERENCE:  
analysis.index observed MLE BAM RBadj  
1 2 0.2 0.2 0.1838 0.1918  
  
Inferences based on Analysis Time Ordering:  
MUE P-value **** CI ****  
1 0.1983 0.01363 (0.041, 0.3548)  
  
Inferences based on Mean Ordering:  
MUE P-value **** CI ****  
1 0.1867 0.01441 (0.0382, 0.3345)  
  
MONITORING BOUNDS:  
Call:  
"(not shown, is (your seqMonitor object)$seqDesignCall)"  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
Two-sided hypothesis test:  
Null hypothesis : Theta = 0 (size = 0.05)  
Alternative hypothesis : Theta > 0.1916 (power = 0.8 )  
  
STOPPING BOUNDARIES: Sample Mean scale  
a b c d  
Time 1 (N= 90) -0.2759 NA NA 0.2759  
Time 2 (N= 130) -0.1891 NA NA 0.1891  
Time 3 (N= 180) -0.1366 -0.1366 0.1366 0.1366
```

The printout of the seqMonitor object indicates that the trial should be stopped with a decision for the upper alternative hypothesis.

Evaluating the Modified Design

By plotting the modified design, you can see that the observed data exceed the revised stopping boundary. On the **Plot** tab of the **Sequential Monitor** dialog,

1. Select **Decision Boundaries**.
2. Select **Overlay Designs**.

Click **Apply**. The result is shown in Figure 10.8.

From the command line, type:

```
> plot(second.analysis)
```

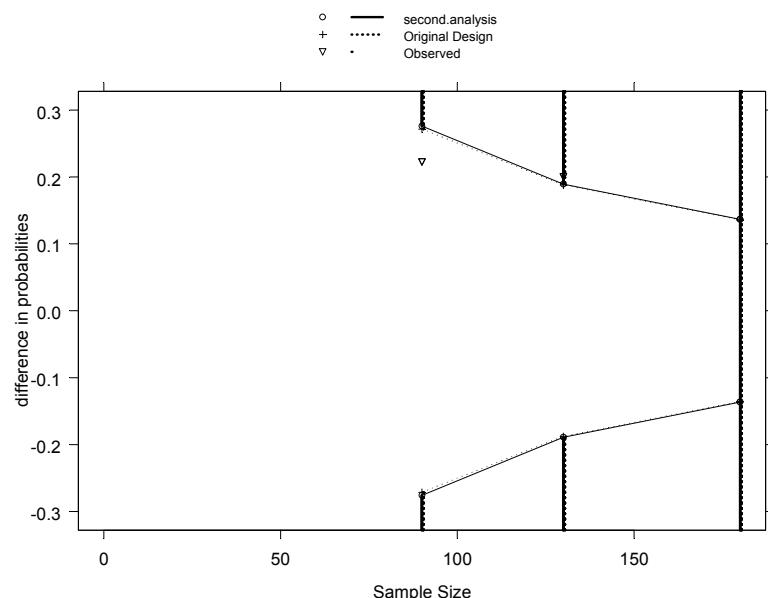


Figure 10.8: The observed data exceed the stopping boundary at the second analysis.

The printed results from the seqMonitor object also present the statistical inferences (p -values, point estimates, and confidence intervals) adjusted for the group sequential stopping rule.

```
INFERENCE:
analysis.index observed MLE      BAM   RBadj
1              2       0.2 0.2 0.1838 0.1918
```

```
Inferences based on Analysis Time Ordering:  
MUE P-value **** CI ****  
1 0.1983 0.01363 (0.041, 0.3548)
```

```
Inferences based on Mean Ordering:  
MUE P-value **** CI ****  
1 0.1867 0.01441 (0.0382, 0.3345)
```

If we focus on the sample mean based inference, which was found by Emerson and Fleming (1990) to have better optimality properties than the other adjusted estimates for a wide variety of settings, the adjusted p -value is .01441, the point estimate of treatment effect is 0.1867, and the 95% confidence interval is .0382 to .3345.

In the original study, the inferences were not adjusted for the interim analyses performed. Instead, the inferences were computed as if the trial was a fixed sample design with a total of 130 subjects. Let's compare these inferences.

From the **Update Design** option under the **SeqTrial** menu, select **aml.fix**. Then, on the **Design** tab,

1. Type 130 in the **N (or N1 ... N1)** field.
2. Type **aml.fix130** in the **Save As** field.

Click **OK**. Then open the **SeqMonitor** dialog, and on the **Design** tab:

1. Type **aml.fix130** in the **Previous Design** field.
2. Type **cr2** in the **Response Vector** field.
3. Type **tx2** in the **Treatment Vector** field.

Click **OK**.

From the command line, type:

```
> aml.fix130 <- update(aml.fix, sample.size=130)  
> seqMonitor(aml.fix130, cr2, tx2)
```

In the fixed sample study, the p -value would have been .012, the point estimate of treatment effect would have been 0.2, and a 95% confidence interval would be .044 to .356. Note that the p -value is too low, the point estimate is too extreme, and the confidence interval is shifted too far away from the null hypothesis relative to the exact inference based on the stopping rule.

PERFORMING A SENSITIVITY ANALYSIS

The clinical trial investigators, guided by the early stopping under the O'Brien–Fleming design, concluded that idarubicin represented a potential improvement in the chemotherapy of acute myelogenous leukemia. The results of this study and three other controlled comparative clinical trials were used in part to support a New Drug Application file in August 1989 with the FDA. During their review of the application, the FDA was concerned that since the interim analyses were not planned in advance, the probability of Type I error may have been inflated above the nominal 0.05 level. To address this concern, we conduct a sensitivity analysis in this section which explores the inferences under designs that vary in the number of early analyses which are performed.

For each set of analysis times, we examine two designs: a “best case” in which the stopping rule was barely exceeded at $N=65$, and a “worst case” in which the stopping rule is nearly as low as the observed rate at $N=45$. The best case finds the O'Brien–Fleming upper boundary which stops at $N=65$ with the highest confidence level. Similarly, the worst case finds the upper boundary which stops at $N=65$ with the lowest confidence level.

Boundary Shift Function

To compute the designs used in the sensitivity analysis, we need to maintain the O'Brien–Fleming boundary relationships, but constrain the stopping rules to pass through the observed data. For general families of group sequential designs, this type of constraint is somewhat difficult to handle, but for the O'Brien–Fleming boundary relationships (and more generally, the Wang & Tsiatis, 1987, relationships), these types of constraints are easy to implement.

In S+SEQTRIAL, the boundaries are usually presented on the scale of the natural parameter θ . The formulas for the boundary relationships on the sample mean scale in the Wang and Tsiatis family are simply:

$$a_{\bar{X}_j} = -G_a \Pi_j^{-P} \quad d_{\bar{X}_j} = G_d \Pi_j^{-P} \quad (10.1)$$

where $P_a = P_d = P$ is the boundary shape parameter, and $G_a = G_d = G$ is some constant chosen to provide the desired Type I error. For the O'Brien–Fleming design, $P_d = 1$. If we want to fix the upper

boundary at a certain point \hat{X}_j at analysis time j , then we need to compute

$$G_d^{\text{new}} = \hat{X}_k \Pi_j^P$$

The new upper boundaries are given by plugging G_d^{new} into Equation (10.1).

Computing the Best and Worst Case Designs

The following function uses Equation (10.1) to create stopping rules that have the boundary shape parameter of the stopping rule specified by `design`, but pass through the point specified by `value` at the j th analysis. The stopping rules are created using the exact constraint capabilities of S+SEQTRIAL.

```
shift.boundary <- function(design, k, value)
{
  ## This function shifts the upper boundaries to be
  ## equal to value at the kth analysis. This function
  ## only works for the special case of Wang & Tsiatis
  ## design. Value is on the sample mean scale.
  P <- design$parameters$P[1]
  N <- design$parameters$sample.size
  PI <- N / N[length(N)]
  G <- value * PI[k] ^ P
  bnd <- seqBoundary(design, display.scale="X")
  bnd[,4] <- PI ^ (-P) * G
  bnd[,1] <- - bnd[,4]
  bnd[,2:3] <- 0
  bnd[length(N),2] <- -G
  bnd[length(N),3] <- G
  update(design, exact.constraint=bnd)
}
```

First compute the “best case” design in which the stopping rule is barely exceeded at $N=65$. This corresponds to fixing the upper boundary at a rate of $13/65 = 0.2$.

```
> aml.best <- shift.boundary(aml.obf, 2, .2)
> print(aml.best)
```

```

Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.6, 0.6), alt.hypothesis = "calculate", ratio =
c(1., 1.), nbr.analyses = 3, sample.size = c(90, 130,
180), test.type = "two.sided", power = 0.8, alpha =
0.025, beta = 0.975, epsilon = c(1., 1.),
exact.constraint = bnd, display.scale = seqScale(
scaleType = "X"))

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0           (size = 0.05 )
Alternative hypothesis : Theta > 0.1991 (power = 0.8 )

STOPPING BOUNDARIES: Sample Mean scale
      a      b      c      d
Time 1 (N= 90) -0.2889     NA     NA 0.2889
Time 2 (N= 130) -0.2000     NA     NA 0.2000
Time 3 (N= 180) -0.1444 -0.1444 0.1444 0.1444

```

Then compute the worst case design in which the stopping rule is nearly as low as the observed rate at $N=45$. This corresponds to fixing the upper boundary at a rate just over $10/45 = 0.2222$.

```

> aml.worst <- shift.boundary(aml.obf, 1, 0.2223)
> print(aml.worst)

Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.6, 0.6), alt.hypothesis = "calculate", ratio =
c(1., 1.), nbr.analyses = 3, sample.size = c(90, 130,
180), test.type = "two.sided", power = 0.8, alpha =
0.025, beta = 0.975, epsilon = c(1., 1.),
exact.constraint = bnd, display.scale = seqScale(
scaleType = "X"))

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0           (size = 0.05 )
Alternative hypothesis : Theta > 0.1653 (power = 0.8 )

```

```
STOPPING BOUNDARIES: Sample Mean scale
      a      b      c      d
Time 1 (N= 90) -0.2223    NA    NA 0.2223
Time 2 (N=130) -0.1539    NA    NA 0.1539
Time 3 (N=180) -0.1112 -0.1112 0.1112 0.1112
```

As you can see, the upper boundaries for these designs meet the desired constraints.

Evaluating the Designs Plot these designs along with the original design and the observed data.

```
> seqPlotBoundary(aml.obf, aml.best, aml.worst,
+     observed=c(10/45, 14/65), N=c(90,130), fixed=F)
```

The result is shown in Figure 10.9.

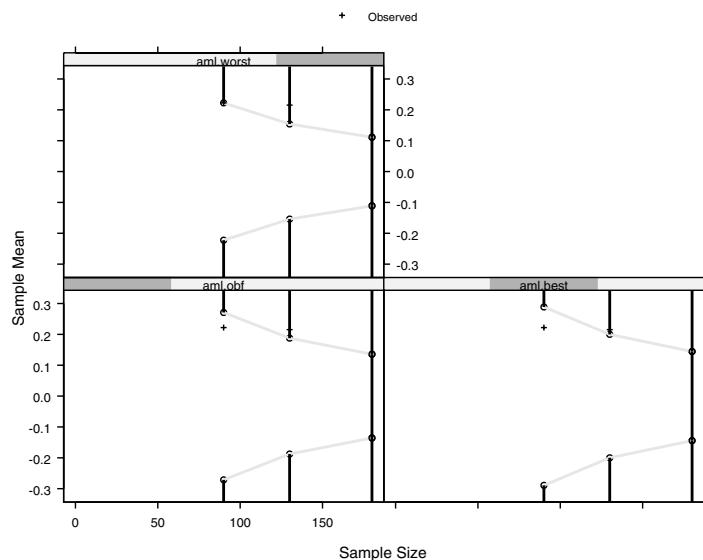


Figure 10.9: The stopping boundaries for “best case” and “worst case” designs, along with the original design and the observed data.

The best case design is very close to the putative stopping rule used for the clinical trial while the worst case design has considerably larger stopping intervals.

Compute the inferences for the best and worst cases with

```
> seqInference(aml.best, 2, .2, "X")
> seqInference(aml.worst, 2, .2, "X")
```

The inferences for these designs are very close to the inferences for the original design.

Similarly, to consider the possibility that the researchers had actually conducted additional interim analyses, you could compute best and worst case designs for stopping rules based on a different schedule of interim analyses:

```
> aml.obf1 <- update(aml.obf, nbr.analyses=4,
+   sample.size= c(50,90,130,180))
>
> aml.best1 <- shift.boundary(aml.obf1, 3, .2)
> print(aml.best1)
> seqInference(aml.best1, 3, .2, "X")
>
> aml.worst1 <- shift.boundary(aml.obf1, 2, .2223)
> print(aml.worst1)
> seqInference(aml.worst1, 3, .2, "X")
```

Or for a Pocock boundary:

```
> aml.poc <- update(aml.obf, P=0.5)
>
> aml.best2 <- shift.boundary(aml.poc, 2, .2)
> print(aml.best2)
> seqInference(aml.best2, 2, .2, "X")
>
> aml.worst2 <- shift.boundary(aml.poc, 1, .2223)
> print(aml.worst2)
> seqInference(aml.worst2, 2, .2, "X")
```

The inferences for all of these designs are very close to the inferences for the original design, though the inferences for the worst case designs vary slightly. For example, as more early interim analyses are added, the p -values increase and the point estimates and confidence intervals are shifted closer to the null hypothesis. However, the inferences for the worst case design are still very close to the inferences for the original design. Hence, we can conclude that statistical inferences are still valid despite the retrospective imposition of a stopping rule.

CASE STUDY #3: UNEXPECTED TOXICITIES

11

Overview	290
Designing a Fixed Sample Test	291
Selecting the Model	291
Specifying the Design	291
Designing Group Sequential Tests	295
Specifying the Designs	295
Evaluating the Designs	298
Data-Imposed Stopping Rules	306
Sensitivity Analysis	307

OVERVIEW

Oral mucositis is a common, dose-limiting toxicity observed with the administration of 5-fluorouracil and low dose leucovorin (5-FU/LV) combination chemotherapy for the treatment of various cancers. In clinical trials testing the ability of granulocyte colony-stimulating factor (G-CSF) to reduce the incidence of febrile leukopenia, several studies showed a decreased incidence of oral mucositis. Thus, a clinical trial was conducted to test whether joint administration of 5-FU/LV and G-CSF might decrease the incidence and/or severity of oral mucositis. Early in the conduct of the trial, however, a high rate of life-threatening leukopenia was noted.

Thirty-five patients with various tumor types were to receive G-CSF during the first cycle of chemotherapy. No formal interim analyses were planned in the original study. After four patients had completed their first course of chemotherapy, the clinical researcher contacted a statistician for help in interpreting the statistical significance of the observation of life-threatening leukopenia in three of those four patients. While this toxicity endpoint was not entirely unforeseen, the fact that three of the first four patients experienced such a toxicity was unexpected. At the time the researcher initially contacted the statistician, an additional patient was currently in the first cycle of his treatment and it was too early to determine whether that patient would experience life-threatening leukopenia. The question was whether additional patients should continue to be accrued to the study, or whether there was already sufficient evidence that the concurrent administration of G-CSF and 5-FU/LV was too toxic.

This case study illustrates that having software to evaluate and compare designs is critical for determining whether a trial should be terminated. Moreover, the type of analysis conducted in this case study can not be entirely automated—it requires a flexible environment such as that offered by S+SEQTRIAL.

DESIGNING A FIXED SAMPLE TEST

One way to answer the researcher's question is to determine what decisions and inferences would have been made for a trial in which the major endpoint was the incidence of life-threatening leukopenia.

Note that although the inferences obtained from a stopping rule imposed retroactively are not strictly valid, the fact that the stopping rule is data-driven has a relatively small impact on the inferences in this case. This issue is explored in more detail later in this chapter.

Selecting the Model

There is a single treatment group and the primary endpoint is toxicity with a binary outcome. As a statistical model, this is a one-sample Binomial hypothesis test. Choose **SeqTrial ► One Sample ► Binomial Proportion** to launch the appropriate S+SEQTRIAL dialog for this trial.

All of the commands needed to run this case study from the command line are contained in a help file. Type:

```
> help(case3.seqtrial)
```

Note: Since this trial deals with such a small number of patients, the exact stopping rules and inferences should be computed using the discrete Binomial distribution, as was done in Emerson (1995). The current version of S+SEQTRIAL only supports the normal approximation to the Binomial. Hence, the analysis presented below is only an approximation. This approximation is, however, close enough to the true Binomial distribution for the conclusions and inferences to be valid. (Compare the results with those presented by Emerson, 1995.)

Specifying the Design

Let's first define a fixed sample design that achieves the desired size and power to satisfy the researcher's concern regarding toxicity. The clinical researcher said that he would tolerate a toxicity rate of up to $\pi = .5$. To test the toxicity rate, form a one-sided hypothesis test $H_0 : \pi \leq \pi_0$ versus $H_1 : \pi \geq \pi_1$. Since the maximal sample size is fixed at 35 patients, we cannot achieve arbitrary Type I error and power constraints. With a fixed sample design, we can fix the Type I error at .025 for $\pi_0 = .2$ and achieve power .975 for an alternative $\pi_1 \approx .5$.

With this choice of Type I error and power, at the conclusion of the study, we either reject H_0 or H_1 . If we reject H_1 , then we can conclude that the toxicity rate is less than .5, and hence, within the bounds specified by the researcher. On the other hand, if we reject H_0 , then the toxicity rate may well be greater than .5.

Let's specify this design.

1. Select the **Min. Difference** option for the computational task. In this case, S+SEQTRIAL computes the alternative for which a given sample size has the desired power; you must specify the sample size and the power.
2. Specify the **Probabilities**. Ensure that the **Significance Level** is set to .025, and the **Power** is set to .975 (the defaults).
3. Set the **Sample Size** to 35.
4. The toxicity rate under the null hypothesis is $\pi_1 = 0.2$, so set the **Null Proportion** field to 0.2.
5. Ensure that the **Variance Method** is set to alternative and that the **Test Type** is set to greater (the defaults).
6. Save the design object under the name `toxic.fix` using the **Save As** field.

The **Design** tab should now look like that shown in Figure 11.1.

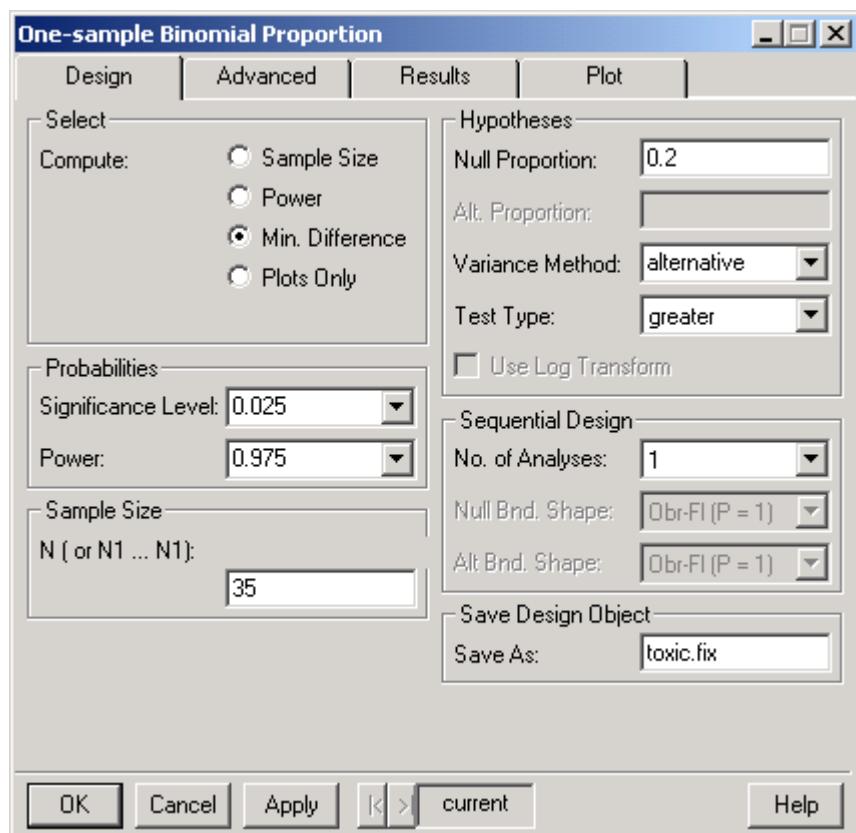


Figure 11.1: *Design* tab for a fixed sample test.

From the command line, the same model can be selected and designed by typing:

```
> toxic.fix <- seqDesign(prob.model = "proportions",
+   arms = 1, null.hypothesis = 0.2,
+   alt.hypothesis="calculate",
+   variance = "alternative",
+   test.type = "greater", sample.size = 35,
+   size = 0.025, power = 0.975)
```

Click **Apply** to create the design object and print out summary information in a **Report** window, as follows.

Chapter 11 Case Study #3: Unexpected Toxicities

```
Call:  
seqDesign(prob.model = "proportions", arms = 1,  
         null.hypothesis = 0.2, variance =  
         "alternative", sample.size = 35, test.type =  
         "greater", size = 0.025, power = 0.975)  
  
PROBABILITY MODEL and HYPOTHESES:  
One arm study of binary response variable  
Theta is event probability  
One-sided hypothesis test of a greater alternative:  
    Null hypothesis : Theta <= 0.2      (size = 0.025 )  
Alternative hypothesis: Theta>=0.5307 (power=0.975)  
(Fixed sample test)  
  
STOPPING BOUNDARIES: Sample Mean scale  
          a      d  
Time 1 (N= 35) 0.3653 0.3653
```

From the command line, you can display the same information about toxic.fix using the print function:

```
> print(toxic.fix)
```

This design achieves the desired size and power to satisfy the researcher's concern regarding toxicity. At the conclusion of the study, if we reject the alternative hypothesis, then we can conclude that toxicity rate is less than 0.5307, and hence, within the bounds specified by the researcher. On the other hand, if we reject the null hypothesis, then the toxicity rate may well be greater than 0.5307.

DESIGNING GROUP SEQUENTIAL TESTS

Since the results of four patients were already available and a fifth patient was under treatment, it makes sense to consider group sequential designs with the first analysis at $N_1 = 5$. The number and timing of additional analyses are then set to maintain a constant rate of monitoring up to the total accrual of $N= 35$ patients—namely, a total of 7 analyses with each analysis occurring after accruing a group of five patients. It remains to find stopping rules which satisfy the above Type I error and sample size constraints.

Specifying the Designs

Let's compare two commonly used designs: the O'Brien–Fleming design (O'Brian & Fleming, 1979) and the Pocock design (Pocock, 1977).

First specify the O'Brien–Fleming design.

1. Select the **Design** tab.
2. Set the **Number of Analyses** to 7. This specifies the total number of analyses (interim plus final). By default, the analyses are evenly spaced according to sample size—in this case, after 5, 10, 15, 20, 25, 30, and 35 subjects have been accumulated.
3. Set **Early Stopping** to **Alternative**.
4. Save the design object under the name **toxic.obf** using the **Save As** field.

Note that this design has the same size and power under the design alternative as the fixed sample design.

From the command line, the same model can be selected and designed by typing:

```
> toxic.obf <- update(toxic.fix, nbr.analyses=7,
+ early.stopping="alternative", P=1)
```

Click **Apply** to create the design object and print out summary information in a report window:

Chapter 11 Case Study #3: Unexpected Toxicities

```
seqDesign(prob.model = "proportions", arms = 1, null.hypothesis =  
 0.2, alt.hypothesis = "calculate", nbr.analyses = 7, sample.size  
 = 35, test.type = "greater", power = 0.975, alpha = 0.025, beta  
 = 0.975, epsilon = c(0., 1.), early.stopping = "alternative",  
 display.scale = seqScale(scaleType = "X"))  
  
PROBABILITY MODEL and HYPOTHESES:  
One arm study of binary response variable  
Theta is event probability  
One-sided hypothesis test of a greater alternative:  
Null hypothesis : Theta <= 0.2      (size = 0.025 )  
Alternative hypothesis : Theta >= 0.5353   (power = 0.975 )  
  
STOPPING BOUNDARIES: Sample Mean scale  
          a      d  
Time 1 (N= 5) -Inf 1.4176  
Time 2 (N= 10) -Inf 0.8088  
Time 3 (N= 15) -Inf 0.6059  
Time 4 (N= 20) -Inf 0.5044  
Time 5 (N= 25) -Inf 0.4435  
Time 6 (N= 30) -Inf 0.4029  
Time 7 (N= 35) 0.3739 0.3739
```

From the command line, display the same information by typing:

```
> print(toxic.obf)
```

Of interest here is the stopping rule for the upper boundary d . By default the boundaries are reported on the Sample Mean scale:

```
STOPPING BOUNDARIES: Sample Mean scale  
          a      d  
Time 1 (N= 5) -Inf 1.4176  
Time 2 (N= 10) -Inf 0.8088  
Time 3 (N= 15) -Inf 0.6059  
Time 4 (N= 20) -Inf 0.5044  
Time 5 (N= 25) -Inf 0.4435  
Time 6 (N= 30) -Inf 0.4029  
Time 7 (N= 35) 0.3739 0.3739
```

Let's examine the boundaries on the partial sum scale, which in this case is equal to the number of observed toxicities.

1. Select the **Results** tab.
2. Change the **Display Scale** field in the **Options** groupbox to **Partial Sum**.
3. Click **Apply**.

The boundaries are now displayed in terms of the number of observed toxicities.

```
STOPPING BOUNDARIES: Cumulative Sum scale
      a      d
Time 1 (N= 5)    -Inf  7.0882
Time 2 (N= 10)   -Inf  8.0882
Time 3 (N= 15)   -Inf  9.0882
Time 4 (N= 20)   -Inf  10.0882
Time 5 (N= 25)  -Inf  11.0882
Time 6 (N= 30)  -Inf  12.0882
Time 7 (N= 35)  13.0882 13.0882
```

From the command line, you can change the display scale using the update function

```
> toxic.obf <- update(toxic.obf, display.scale="S")
```

then reprint the results by typing

```
> print(toxic.obf)
```

These stopping boundaries are very conservative at the early analyses. In fact, it is not possible to stop at the first analysis at all with these rules. The first analysis occurs after 5 subjects have been accrued, but the stopping boundary requires 8 toxicities to stop. At the second boundary 9 of the 10 subjects must show toxicity for the trial to stop.

Now let's create the Pocock design.

1. Select the **Design** tab.
2. Set the **Boundary Shape** field to Pocock ($P=.5$).
3. Save the new design under the name **toxic.poc** using the **Save As** field.
4. Click **Apply**.

From the command line, the same model can be selected and designed by typing:

```
> toxic.poc <- update(toxic.obf, P=0.5)
```

Results can be printed by typing:

```
> print(toxic.poc)
```

The Pocock boundaries are much less conservative at the early analyses.

```
STOPPING BOUNDARIES: Cumulative Sum scale
      a      d
Time 1 (N= 5)   -Inf  3.7579
Time 2 (N= 10)   -Inf  5.9002
Time 3 (N= 15)   -Inf  7.7768
Time 4 (N= 20)   -Inf  9.5157
Time 5 (N= 25)   -Inf 11.1668
Time 6 (N= 30)   -Inf 12.7554
Time 7 (N= 35) 14.2966 14.2966
```

This design stops at the first analysis if 4 of the 5 subjects exhibit toxicity, and at the second analysis if 6 of the 10 subjects do.

Evaluating the Designs

After accruing $N_k = (5, 10, 15, 20, 25, 30, 35)$ patients, the O'Brien–Fleming rules imply stopping when the number of toxicities is $S_k = (8, 9, 10, 11, 12, 13, 14)$ or greater. The corresponding Pocock stopping rules are $S_k = (4, 6, 8, 10, 12, 13, 15)$ or greater. Let's compare these stopping rules graphically.

1. Select the **Plot** tab.
2. Select the **Decision Boundaries** plot type.
3. Set the **Display Scale** to **Partial Sum**.
4. In the **Designs to Plot** listbox, click the **Refresh** button, then select design **toxic.obf**. This design will be compared to the current design (**toxic.poc**).
5. Select the **Overlay Designs** option.
6. Click **Apply**.

The result is shown in Figure 11.2.

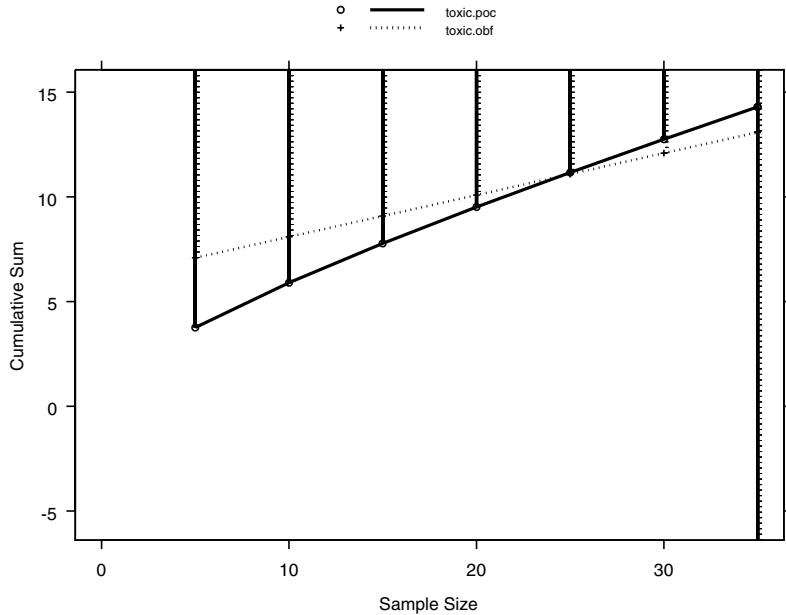


Figure 11.2: A comparison of stopping rules based on an O'Brien–Fleming design `toxic.obf` and a Pocock design `toxic.poc`.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(toxic.poc, toxic.obf,
+     superpose.design=T, display.scale="S")
```

The early conservatism of the O'Brien–Fleming rules is clear: it is not possible to stop at the first analysis. By contrast, the Pocock design stops if four or five toxicities are observed at the first analysis.

Now let's compare the stopping probabilities for the two designs.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Stopping Probability** plot type.
3. Deselect the **Overlay Designs** option.
4. Click **Apply**.

The result, shown in Figure 11.3, further highlights the early conservatism of the O'Brien–Fleming relationship.

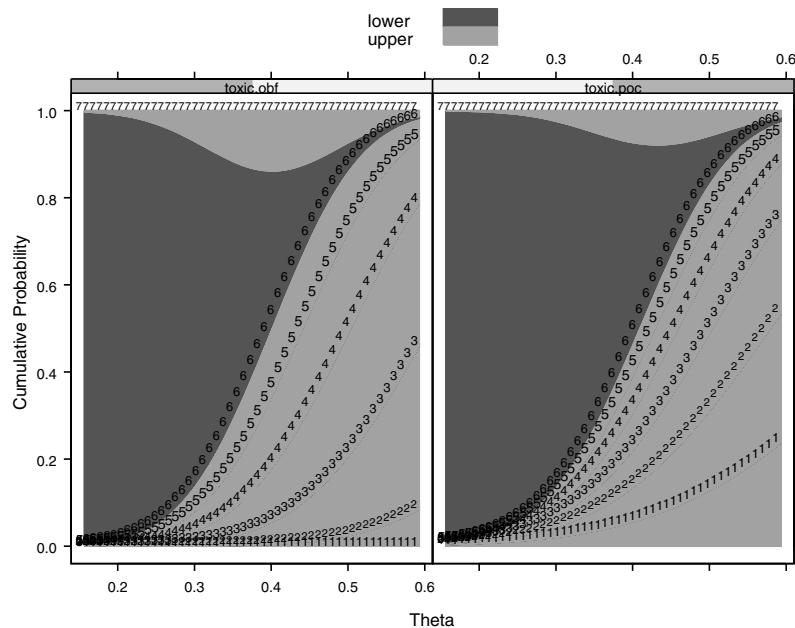


Figure 11.3: The stopping probabilities for the O'Brien–Fleming and Pocock designs.

From the command line, the same plot can be created by typing:

```
> seqPlotStopProb(toxic.poc,toxic.obf)
```

The numbers indicate the probability of stopping by the k th analysis. This one-sided design stops early only for rejection of H_0 . The Pocock design has a much higher probability of stopping early. For example, compare the lines marked 1 and 2 on the two plots.

To examine the efficiency of the designs with regards to the number of patients that need to be accrued, let's also plot the ASN and sample size quantile curves.

1. Deselect the **Stopping Probability** plot type.
2. Select the **Average Sample Number** plot type.
3. Select the **Overlay Designs** option.
4. Click **Apply**.

The result is shown in Figure 11.4.

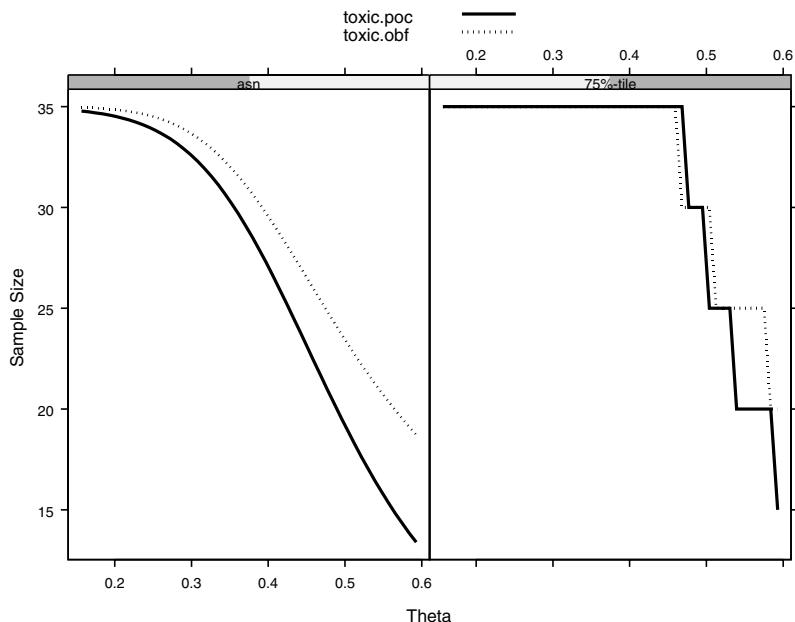


Figure 11.4: The average sample size (ASN) curve and the 75th %tile curve for the O'Brien–Fleming and Pocock designs.

From the command line, the same plot can be created by typing:

```
> seqPlotASN(toxic.poc, toxic.obf, fixed=F)
```

On average, the Pocock design uses fewer patients than the O'Brien–Fleming design.

In comparing designs, it is also useful to examine the inferences which would be made at critical decision points.

1. Deselect the **Average Sample Number** plot type.
2. Select the **Results** tab.
3. Deselect the **Short Output** option.
4. Select the **Summary Tables** option.
5. Click **Apply**.

From the command line, the same information can be printed by typing:

```
> summary(toxic.poc)
```

The inferences at the “d” boundary for the first and final analyses are shown below.

Ordering		*** d Boundary ***
Time 1	Boundary	3.758
	MLE	0.752
	BAM	0.686
	RBadj	0.752
Mean	MUE	0.694
Mean	P-value	0.007
Mean 95% Conf Int		(0.313, 1.027)
Time	MUE	0.752
Time	P-value	0.006
Time 95% Conf Int		(0.317, 1.187)
.		
.		
.		
Time 7	Boundary	14.297
	MLE	0.408
	BAM	0.367
	RBadj	0.349
Mean	MUE	0.390
Mean	P-value	0.025
Mean 95% Conf Int		(0.200, 0.561)
Time	MUE	0.390
Time	P-value	0.025
Time 95% Conf Int		(0.200, 0.561)

Now print the same information for design toxic.obf.

1. Choose **SeqTrial ▶ Update Design** from the S-PLUS menu bar.
2. Select **toxic.obf** from the listbox and click **OK**.
3. Select the **Results** tab.
4. Deselect the **Short Output** option.
5. Select the **Summary Tables** option.
6. Click **Apply**.

From the command line, the same information can be printed by typing:

```
> summary(toxic.obf)
```

The inferences at the “d” boundary for the second and final analyses are shown below. (No early stopping is possible at the first analysis for this design, so the inference that would be made at that boundary is nonsensical.)

	Ordering	*** d Boundary ***
Time 2	Boundary	8.088
	MLE	0.809
	BAM	0.782
	RBadj	0.809
Mean	MUE	0.775
Mean	P-value	0.000
Mean 95% Conf Int		(0.493, 1.040)
Time	MUE	0.809
Time	P-value	0.000
Time 95% Conf Int		(0.500, 1.118)
.		
.		
.		
Time 7	Boundary	13.088
	MLE	0.374
	BAM	0.358
	RBadj	0.355
Mean	MUE	0.368
Mean	P-value	0.025
Mean 95% Conf Int		(0.200, 0.535)
Time	MUE	0.368
Time	P-value	0.025
Time 95% Conf Int		(0.200, 0.535)

The inferences for the two designs are compared in Table 11.1.

Table 11.1: Inferences at the “ d ” boundary that would be made for O’Brien–Fleming and Pocock designs.

	O’Brien–Fleming	Pocock
First possible stopping		
S_M'/N_M	9/10	4/5
p -value	0.000	0.007
Point estimate (MLE)	0.809	0.752
95% CI	(0.493, 1.040)	(0.313, 1.027)
Rejection of H_0 at final analysis time		
S_M'/N_M	14/35	15/35
p -value	0.025	0.025
Point estimate	0.374	0.408
95% CI	(0.200, 0.535)	(0.200, 0.561)

The conservatism of the O’Brien–Fleming boundary relationships is again apparent at the earliest stopping time. On the other hand, the inferences at the boundary for rejection of H_0 at the last analysis is not markedly different between the two designs.

Based on the above comparisons, the Pocock boundary relationships were chosen in Emerson (1995). The conservatism of the O’Brien–Fleming rule is inappropriate for this trial: you would not want to continue a trial merely to establish an excessive rate of toxicity. When the results of the fifth patient were available, it was found that the patient had also experienced life-threatening

leukopenia. Thus, according to the Pocock stopping rule, the occurrence of four toxicities in the first five patients was sufficient evidence to terminate the trial with the conclusion that concurrent administration of G-CSF with 5-FU/LV tended to produce unacceptably high rates of life-threatening leukopenia.

DATA-IMPOSED STOPPING RULES

The analysis of the previous section is technically valid only for the planning of a sequential trial prior to accrual of any patients. In this trial, the stopping rule was imposed after the results for the first four patients was available. We now explore the impact that such retroactive imposition of a stopping rule might have had on the statistical inference from the study.

The data-driven interim analyses is modeled as follows. Let $Y_1 \sim B(4, \pi)$ be the binomial random variable measuring the number of patients with life-threatening leukopenia observed among the first four patients accrued to the trial. Assume there is some critical value C such that the observation $Y_1 \geq C$ would cause the researchers to impose a formal stopping rule, which in this case would be the Pocock design described in the previous section. If $Y_1 < C$, the study would be carried out as a fixed sample clinical trial as originally planned.

The power and size are given by

$$Pr(\text{reject } H_0; \pi) =$$

$$\sum_{y=0}^4 Pr(\text{reject } H_0 | Y_1 = y; \pi) \cdot Pr(Y_1 = y; \pi) \quad (11.1)$$

For $y < C$, then

$$Pr(\text{reject } H_0 | Y_1 = y; \pi) = Pr(Y_2 = 12 - y; \pi)$$

where $Y_2 \sim B(31, \pi)$ is a binomial random variable counting the number of toxicities in the remaining 31 patients.

For $y \geq C$, then $Pr(\text{reject } H_0 | Y_1 = y; \pi)$ can be computed as the size of a group sequential test defined for the last 31 patients. This test is merely a shift of the stopping boundaries for the original test derived for $N = 35$ patients. The shifted stopping rule is defined by $N_k^* = N_k - 4$, $a_k^* = a_k - y$, and $d_k^* = d_k - y$.

Sensitivity Analysis

We can now define a special function `cond.size` that computes Equation (11.1). This function takes as arguments a design (`des`) and a hypothesized value of the treatment effect (`theta`).

```
cond.size <- function(des, theta)
{
  if(length(theta) != 1) stop("theta must be scalar")
  bnd <- seqBoundary(des, scale="S")
  N <- sampleSize(des)-4
  ## compute P(reject H(0) | Y=y, C=c)
  ## rows    correspond to y = 0:4
  ## columns correspond to C = 0:5
  cond.size <- matrix(0, nrow=5, ncol=6)
  for(y in 0:3){
    cond.size[y+1, (y+2):6] <- 1-pbinom(12-y, size=31,
      prob=theta)
    new <- update(des, exact.constraint=bnd-y, sample.size=N)
    cond.size[y+1, 1:(y+1)] <- seqOperatingChar(new,
      theta)[,"power.upper"]
  }
  cond.size[5,1:5] <- 1
  cond.size[5,6] <- 1-pbinom(8, size=31,
    prob=theta)
  ## multiply by P(Y=y) and
  ## sum over y to get conditional size (power)
  dimnames(cond.size) <- list(rep("",5), paste("C=",0:5,sep=""))
  colSums(cond.size * dbinom(0:4, size=4, prob=theta))
}
```

Using this function, we can explore the sensitivity of the inference to the value of the unknown critical value C . Let's apply it to compute the adjusted size at the null hypothesis ($\pi_0 = .2$):

```
> round(cond.size(toxic.poc, 0.2), 4)
   C=0     C=1     C=2     C=3     C=4     C=5
0.0207 0.0215 0.0219 0.0198 0.0155 0.0142
```

and the adjusted power at the alternative ($\pi_1 = .5$):

```
> round(cond.size(toxic.poc, 0.5), 4)
   C=0     C=1     C=2     C=3     C=4     C=5
0.8947 0.9047 0.9305 0.9513 0.9556 0.9552
```

The size is relatively insensitive to the value of C , and the loss of power is in keeping with a Pocock design in which the sample size is not increased. Hence, we can conclude that data-driven imposition of the stopping rule has only a small impact on the analysis presented in the previous section.

CASE STUDY #4: EFFICIENCY CONCERNS

12

Overview	310
Initial Protocol	311
Selecting the Model	311
Specifying the Design	311
Sponsor Safety Boundary	316
Specifying the Design	316
Evaluating the Design	319
DSMB Safety Boundary	323
Specifying the Design	323
Evaluating the Design	325
DSMB Efficiency Boundary	329
Specifying the Design	329
Evaluating the Design	330
Sponsor Futility Boundaries	335
Specifying the Designs	335
Evaluating the Designs	336
Advanced Boundary Scales	341
Conditional Futility Scale	341
Predictive Futility Scale	343
Bayesian Scale	344

OVERVIEW

A pharmaceutical company conducted a clinical trial for a new biologic agent in the treatment of severe sepsis. Two previous Phase III trials of this agent provided promising but inconclusive results suggesting the treatment might be effective in certain subsets of patients with severe sepsis. In order to follow-up on these results, a large Phase III study was conducted in a more precisely defined population of patients.

The design of this study involved interactions between the sponsor, the FDA, and the DSMB. Participants in the design had a wide range of backgrounds, including statisticians, medical researchers, government health administrators, and pharmaceutical business executives. The key point of this case study is that to meet the concerns of the different groups and individuals involved in the design, it was important to evaluate and compare a range of stopping rules. The comparisons consisted of power curves, ASN curves, and the conditional and predictive probability that a statistically significant result would eventually be obtained.

INITIAL PROTOCOL

The sponsors of the study initially proposed a large, fixed sample clinical trial. The primary endpoint for the trial was 14-day mortality. Little other information was collected. This strategy was acceptable to the FDA due to the existence of safety data from two previous studies, and the fact that the treatment itself was administered only on the first two days of the observation period.

Selecting the Model

The proposed trial randomly assigned subjects with equal probability to a treatment or placebo control group. The primary endpoint of the trial was the 14-day mortality rate. As a statistical model, this can be cast as a two-sample Binomial hypothesis test. Choose **SeqTrial ► Two Samples ► Binomial Proportions** to launch the appropriate S+SEQTRIAL dialog for this trial.

All of the commands needed to run this case study from the command line are contained in a help file. Type:

```
> help(case4.seqtrial)
```

Specifying the Design

Let's design the fixed sample hypothesis test.

1. Ensure that the **Sample Size** radio button in the **Compute** group box is selected (the default).
2. Specify the **Probabilities**. In addition to their interest in the treatment effect in the overall study population, the sponsors wanted to examine the treatment effect in a subgroup restricted to those patients with severe multiorgan failure. In order to control the overall experimentwise Type I error at .05, it was determined that the test of the primary endpoint should be carried out at a .045 level of significance, and the test in the subgroup analysis should be carried out at a .01 level of significance. (Group sequential methods were used to derive these alpha-levels. In essence, the subgroup was treated much like a first analysis, and the analysis of the primary endpoint using the entire study population was treated like a final analysis. Further discussion of this point is beyond the scope of this presentation.) Set the **Significance Level** to **.045** and the **Power** to **.90**.

3. Specify the null hypothesis. Based on prior experience, it was estimated that the patients receiving the placebo would have 30% mortality during the first 14 days. Enter **0.30,0.30** in the **Null Proportions** field, indicating the outcome on the treatment arm and the control arm under the null hypothesis.
4. Specify the alternative hypothesis. The researchers hoped that the treatment would reduce mortality to 23%, or an absolute decrease of .07 in the proportion of patients dying in the first 14 days. Enter **0.23,0.30** in the **Alt Proportions** field, indicating the outcome on the treatment arm and the control arm under the alternative hypothesis.
5. Ensure that the **Variance Method** is set to **alternative** (the default), signifying that the power calculation should be performed assuming the variability specified by the mean-variance relationship under the alternative hypothesis.
6. Set the **Test Type** to **two.sided** for a two-sided hypothesis test.
7. This is a fixed sample design, so ignore the **Sequential Design** group box. Later we'll modify our design to include interim analyses.
8. Save the design object under the name **sepsis.fix** using the **Save As** field.

The **Design** tab should now look like that shown in Figure 12.1.

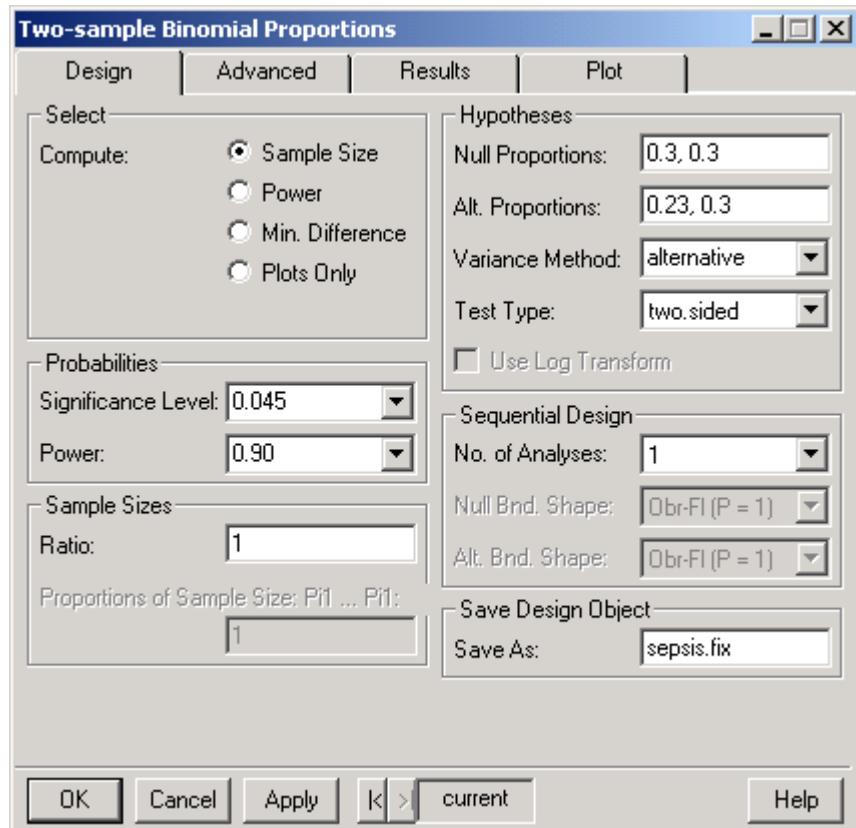


Figure 12.1: *Sample S+SEQTRIAL* dialog for a fixed sample design.

From the command line, the same model can be selected and designed by typing:

```
> sepsis.fix <- seqDesign(prob.model="proportions",
+    arms=2, size=.045, power=.9,
+    null.hypothesis= c(.30, .30),
+    alt.hypothesis=c(.23, .30),
+    test.type="two.sided")
```

Click **Apply** to create the design object and print out summary information in a report window, as follows.

Chapter 12 Case Study #4: Efficiency Concerns

```
Call:  
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis  
= c(0.3, 0.3), alt.hypothesis = c(0.23, 0.3), test.type  
= "two.sided", size = 0.045, power = 0.9)  
  
PROBABILITY MODEL and HYPOTHESES:  
Two arm study of binary response variable  
Theta is difference in probabilities (Treatment - Comparison)  
Two-sided hypothesis test:  
Null hypothesis : Theta = 0 (size = 0.045 )  
Alternative hypothesis : Theta < -0.07 (power = 0.9 )  
(Fixed sample test)  
  
STOPPING BOUNDARIES: Sample Mean scale  
a b c d  
Time 1 (N= 1706.27) -0.0427 -0.0427 0.0427 0.0427
```

From the command line, you can display the same information about `sepsis.fix` using the `print` function:

```
> print(sepsis.fix)
```

This design has a sample size of 1706 (853 per arm). The boundaries on the Sample Mean scale are -.0427 and .0427. These are the critical values for rejecting the null hypothesis, in terms of the observed difference between the mortality rate on the placebo arm minus the mortality rate on the treatment arm.

Based on these calculations, the sponsors decided on a fixed sample study in which 1700 subjects would be accrued (850 on each arm).

1. Select the **Min. Difference** radio button in the **Compute** group box. In this case, you must specify the number of subjects, and S+SEQTRIAL will calculate the alternative hypothesis.
2. Set **N** in the **Sample Sizes** groupbox to **1700**, and ensure that the **Ratio** is set to 1 (the default), indicating that randomization will be in equal proportions to the treatment and control groups.
3. Set the **Test Type** to **two.sided (lower)** for a two-sided hypothesis test. The **lower** option indicates that S+SEQTRIAL should calculate the alternative that the lower hypothesis test can detect.

From the command line, the same model can be selected and designed by typing:

```
> sepsis.fix <- update(sepsis.fix, sample.size=1700,
+ alt.hypothesis="lower")
```

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.3, 0.3), alt.hypothesis = "lower", sample.size =
1700, test.type = "two.sided", size = 0.045, power = 0.9)

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0           (size = 0.045 )
Alternative hypothesis: Theta<-0.07012 (power=0.9 )
(Fixed sample test)

STOPPING BOUNDARIES: Sample Mean scale
      a      b      c      d
Time 1 (N= 1700) -0.0428 -0.0428 0.0428 0.0428
```

From the command line, you can display the same information about `sepsis.fix` using the `print` function:

```
> print(sepsis.fix)
```

This design has a power of 0.9 to detect a minimum treatment effect of -0.07012. Thus, reducing the sample size by 6 subjects had only a negligible effect on the alternative hypothesis.

SPONSOR SAFETY BOUNDARY

In the initial protocol design, no interim monitoring of the data was planned. It was projected that subjects would be accrued over an approximately 15 month period. In reviewing the proposed protocol, however, the FDA requested that the sponsors devise a monitoring plan in order to ensure patient safety.

Specifying the Design

The sponsors then introduced a tentative stopping rule based on a single interim analysis halfway through the study and an *ad hoc* stopping boundary designed to detect a harmful treatment.

S+SEQTRIAL allows you to specify such *ad hoc* stopping boundary by forming a matrix of boundary values. For a design with J analyses, the matrix takes the form

$$\begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \dots & \dots & \dots & \dots \\ a_j & b_j & c_j & d_j \end{bmatrix}$$

Given an observed test statistic T_j , the test stops at the j th analysis as follows:

$$H_- \text{ if } a_j < T_j$$

$$H_0 \text{ if } b_j < T_j < c_j$$

$$H_+ \text{ if } T_j < d_j$$

where the boundaries $a-d$ must be chosen to be appropriate for the exact form of the test statistic T .

For more information on constrained boundaries, see page 158.

First let's define a group sequential design with two equally spaced analyses.

1. Click the **Interim Analyses** checkbox to specify a group sequential design.
2. Change the **Number of Analyses** to 2. The **Boundary Shape** fields can be ignored, because we are going to specify an *ad hoc* design that does not belong to any particular family.
3. Save the design object under the name **sepsis.safety** using the **Save As** field.

Now let's add an *ad hoc* stopping rule. The sponsors found it convenient to work with normalized Z-values. The boundary was defined so early stopping would occur if a test for differences in mortality rejected the null hypothesis of no harm due to treatment in a one-sided, level .05 test. That is, the upper boundary corresponded to a Z statistic of 1.6450 at the interim analysis. At the final analysis, the test continued to use the boundary for a two-sided, level .045 test, corresponding to Z statistics of -2.0047 and 2.0047.

1. Click the **Advanced** tab.
2. In the **Boundary Matrices** groupbox, select **Z-Statistic** from the **Exact Constraint** listbox.
3. A spreadsheet appears with rows for each of the analysis times and columns for each of the four boundaries. S+SEQTRIAL considers designs that have up to four boundaries: a lower "a" boundary that rejects the null hypothesis for a lower alternative, an inner "b" boundary that rejects the lower alternative, an inner "c" boundary that rejects the upper alternative, and an upper "d" boundary that rejects the null hypothesis for an upper alternative. Enter stopping values of **-Inf, 0, 0, 1.6450** for Time1. Setting b_1 and c_1 to 0 and $a_1 = -\text{Inf}$ = $-\infty$ avoids stopping for the null and lower hypotheses respectively at the interim analysis. Enter stopping values of **-2.0047** for the first two boundaries for Time2, and **2.0047** for the last two boundaries.

The **Exact Constraint** matrix should now look like that shown in Figure 12.2.

		1	2	3	4
		a	b	c	d
1	Z.Time1	-Inf	0	0	1.645
2	Z.Time2	-2.0047	-2.0047	2.0047	2.0047
3					

Figure 12.2: An ad hoc stopping boundary specified on the Z-Statistic scale using an exact constraint matrix.

From the command line, the same model can be selected and designed by typing:

```
> bound <- seqBoundary(scale = "Z",
+   rbind(c(-Inf, 0, 0, 1.645),
+   c(-2.0047,-2.0047,2.0047,2.0047)))
> sepsis.safety <- update(sepsis.fix, nbr.analyses=2,
+   exact.constraint=bound)
```

Click **Apply** to create the design object and print out summary information in a **Report** window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.3, 0.3), alt.hypothesis = "lower", nbr.analyses
= 2, sample.size = 1700, test.type = "two.sided", size
= 0.045, power = 0.9, exact.constraint = bound)

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
Two-sided hypothesis test:
Null hypothesis : Theta = 0           (size = 0.045 )
Alternative hypothesis : Theta < -0.07012 (power = 0.9 )

STOPPING BOUNDARIES: Sample Mean scale
                  a      b      c      d
Time 1 (N= 850) -Inf     NA     NA 0.0496
Time 2 (N= 1700) -0.0428 -0.0428 0.0428 0.0428
```

From the command line, you can display the same information about `sepsis.safety` using the `print` function:

```
> print(sepsis.safety)
```

Adding the interim analysis has not changed the alternative hypothesis (-0.07012).

Note that although we specified the constraints with normalized Z-values, the printed output still reports the critical boundaries on the Sample Mean scale. We could use the **Display Scale** field on the **Results** tab to change the output scale, but for now let's leave it the way it is.

Evaluating the Design

Let's compare the average sample size curves for a range of possible treatment effects.

1. Select the **Plot** tab.
2. Select the **Average Sample Number** plot.
3. In the **Designs to Plot** listbox, click the **Refresh** button, then select the initial protocol design `sepsis.fix`.
4. Select the **Overlay Designs** option in the **Options** groupbox.

Click **Apply**.

From the command line, you can plot the average sample size curve using the `seqPlotASN` function:

```
> seqPlotASN(sepsis.safety, sepsis.fix,
+             superpose.design=T, fixed=F)
```

The result is displayed in Figure 12.3.

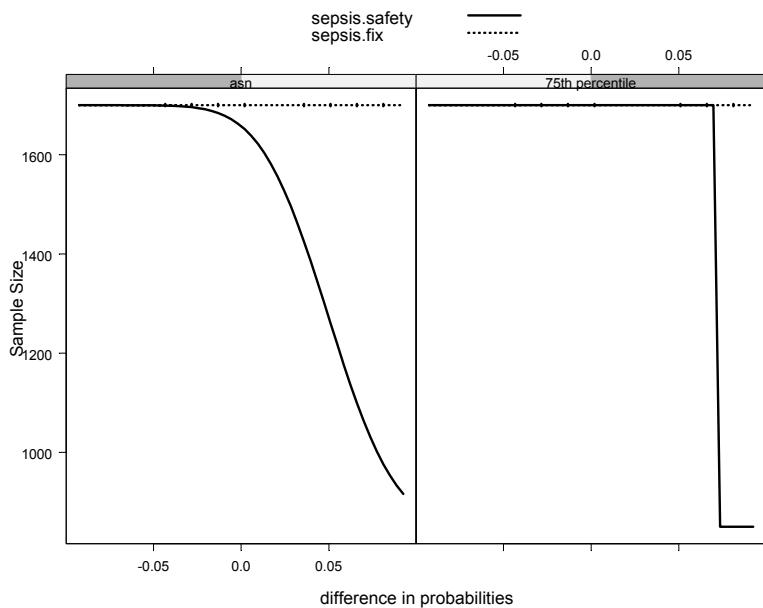


Figure 12.3: The average sample size (ASN) curve for designs `sepsis.safety` and `sepsis.fix`.

The sequential design has a lower average sample size only when treatment is pretty harmful (that is, when treatment minus control is greater than zero).

Now compare the power curves:

1. Deselect the **Average Sample Number** plot.
2. Select the **Power Curve** plot.

Click **Apply**. Figure 12.4 shows the result.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(sepsis.safety, sepsis.fix)
```

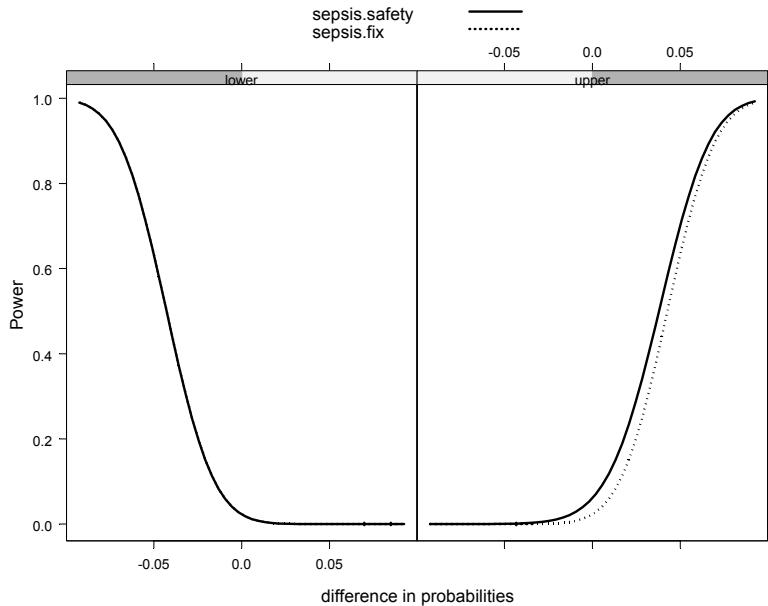


Figure 12.4: The power curves for designs `sepsis.safety` and `sepsis.fix`.

There appears to be a small change in the upper power. It is easier to compare the power curves by plotting the difference between them.

1. Deselect the **Overlay Designs** and **Power Curve** options.
2. Select the **Power Curve vs. Reference** plot type.
3. Set the **Reference Design** to `sepsis.fix`.

Click **Apply**. The result in this case is shown in Figure 12.5.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(sepsis.safety, reference=sepsis.fix)
```

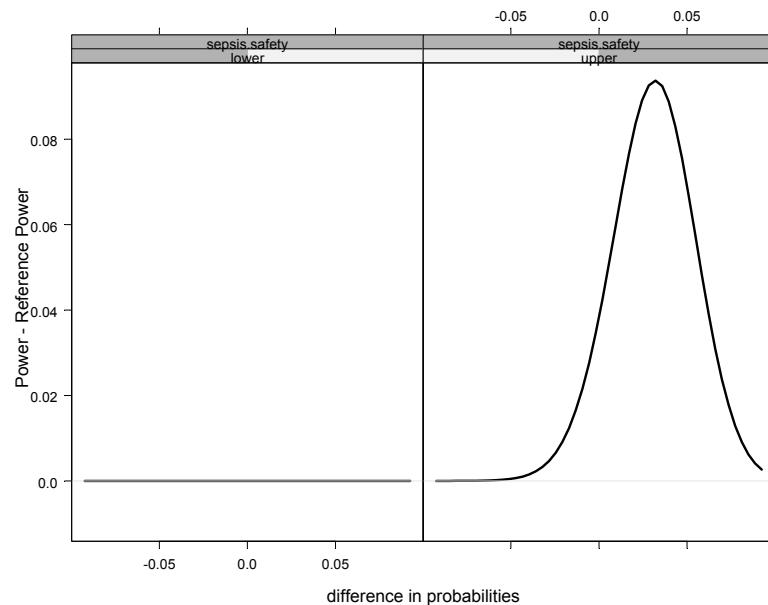


Figure 12.5: The difference curve created by subtracting the fixed design `sepsis.fix` from the group sequential design `sepsis.safety`.

There is no change in the power of the lower hypothesis test to detect a benefit of the new treatment, but there is a small change in the power of the upper hypothesis test (less than 1%) to detect a harmful effect from the new treatment.

DSMB SAFETY BOUNDARY

The ad hoc stopping rule was reviewed by the Data Safety Monitoring Board (DSMB) when it was organized near the start of the trial. The DSMB suggested that in a placebo-controlled study, it was not appropriate to continue the trial just to establish that the treatment was worse than placebo. Instead the safety boundary should be based on a one-sided, level 0.0225 test. The DSMB therefore considered a stopping rule based on a one-sided test with an O'Brien-Fleming upper boundary, and no early stopping for efficacy (no lower boundary). Our evaluation of this new stopping rule will consider up to 4 analyses, though at this stage only a single interim analysis was planned.

Specifying the Design To specify this design, modify the upper boundary as follows:

1. Deselect the **Power Curve vs. Reference** plot type.
2. Select the **Advanced** tab.
3. Set the **Exact Constraint** matrix to **None**.
4. Select the **Design** tab.
5. Set the **Significance Level** of the test to **0.0225**.
6. Set the **Test Type** to **less**, indicating a one-sided test in which the difference in means is less under the alternative hypothesis (-0.07) than under the null hypothesis (0.0).
7. Set the **Alt Bnd. Shape** field to **No Early Stopping**, indicating that early stopping is possible only under the null hypothesis.
8. Ensure that the **Null Bnd. Shape** parameter P is set to **Obr-Fl ($P=1$)** (the default), which corresponds to an O'Brien-Fleming boundary relationship.
9. Save the design object under the name **sepsis.dsmb** using the **Save As** field.

From the command line, the same model can be selected and designed by typing:

```
> sepsis.dsmb <- update(sepsis.safety, size=0.0225,
+   test.type="less", nbr.analyses=4, P=1,
+   early.stopping="null", exact.constraint=,
+   epsilon=,alpha=)
```

Removing the epsilon and alpha arguments is not necessary if sepsis.safety was created from the command line, but is if it was created from the menus.

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.3, 0.3), alt.hypothesis = "lower", nbr.analyses
= 4, sample.size = 1700, test.type = "less", size =
0.0225, power = 0.9, early.stopping = "null", P = 1)

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
Null hypothesis : Theta >= 0           (size = 0.0225 )
Alternative hypothesis : Theta <= -0.07063    (power = 0.9 )

STOPPING BOUNDARIES: Sample Mean scale
                     a      d
Time 1 (N= 425)    -Inf  0.0900
Time 2 (N= 850)    -Inf  0.0018
Time 3 (N= 1275)   -Inf -0.0276
Time 4 (N= 1700)  -0.0423 -0.0423
```

From the command line, you can display the same information about sepsis.dsmb using the **print** function:

```
> print(sepsis.dsmb)
```

This design has a power of 0.9 to detect a minimum difference of -0.07063.

Evaluating the Design

Let's plot the two sequential designs.

1. Select the **Plot** tab and choose the **Decision Boundaries** plot type.
2. In the **Designs to Plot** listbox, click the **Refresh** and **Deselect All** buttons, then select **sepsis.safety** to compare with the current design (**sepsis.dsmb**).
3. In the **Options** groupbox, choose the **Overlay Designs** option.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(sepsis.dsmb, sepsis.safety,
+     superpose.design=T)
```

The result is shown in Figure 12.6.

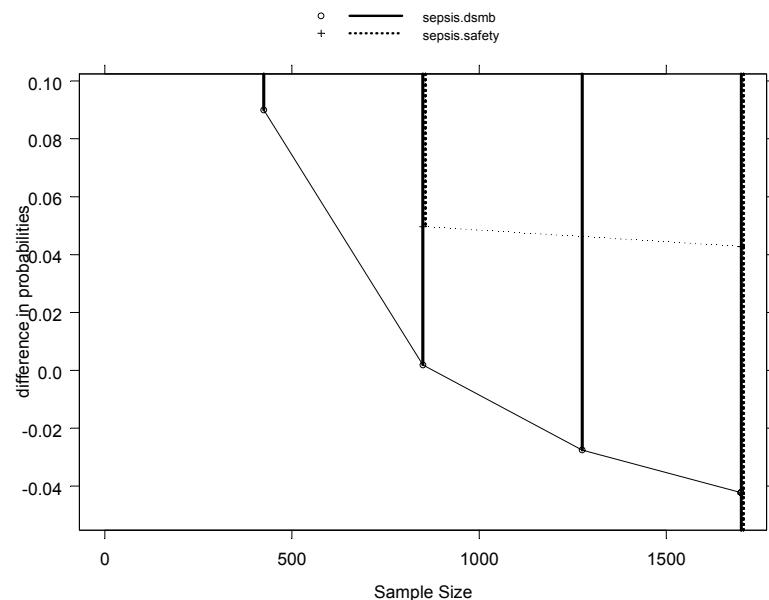


Figure 12.6: A comparison of stopping rules for designs **sepsis.safety** and **sepsis.dsmb**.

Chapter 12 Case Study #4: Efficiency Concerns

Design `sepsis.dsmb` has a much less conservative upper boundary.

Now compare the power curves.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Power Curve** plot type.
3. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotPower(sepsis.dsmb, sepsis.safety)
```

The result is shown in Figure 12.7.

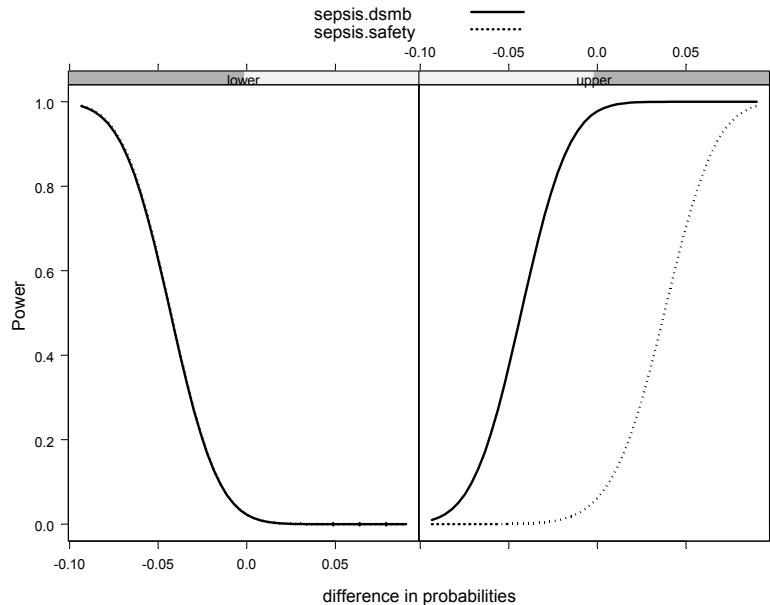


Figure 12.7: The power curves for designs `sepsis.safety` and `sepsis.dsmb`.

There is no change in the lower power, but a large change in the upper power.

Now compare the sample size distributions for these designs.

1. Deselect the **Power Curve** plot type.
2. Select the **Average Sample Number** plot type.
3. In the **Designs To Plot** group box, select **sepsis.fix** in addition to **sepsis.safety**.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotASN(sepsis.dsmb, sepsis.safety, sepsis.fix,
+           fixed=F)
```

The result is shown in Figure 12.8.

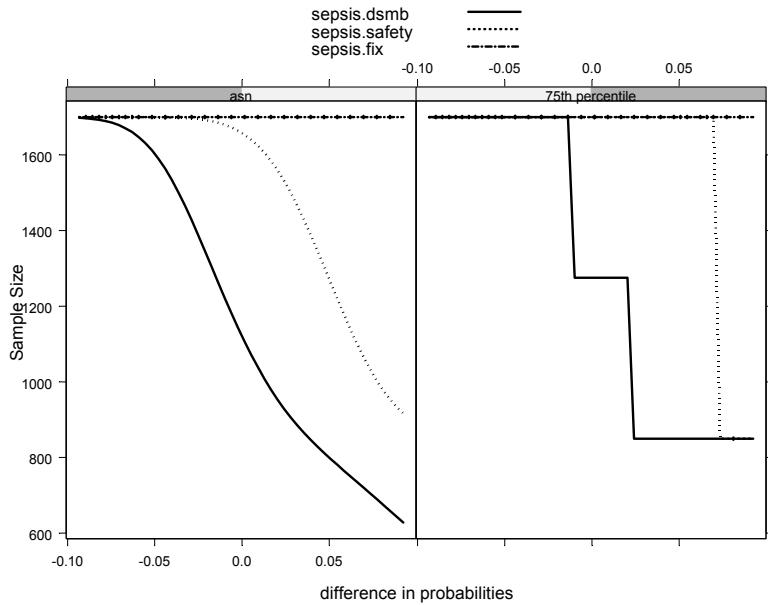


Figure 12.8: The average sample size (ASN) curve and the 75th %tile curve for designs **sepsis.safety** and **sepsis.dsmb**.

Chapter 12 Case Study #4: Efficiency Concerns

Design sepsis.dsmb has a lower average sample size when the treatment is harmful, or even minimally beneficial, indicating good futility behavior.

DSMB EFFICIENCY BOUNDARY

The DSMB also wanted to consider the possibility of early stopping for efficacy. Therefore, a lower boundary was added, also using an O'Brien-Fleming boundary relationship. The resulting test was a one-sided symmetric design as described by Emerson and Fleming (1989).

Specifying the Design

1. Deselect the **Average Sample Number** plot type.
2. Select the **Design** tab.
3. Set the **Early Stopping** field to **Both**.
4. Save the design object under the name **sepsis.dsmb2** using the **Save As** field.

From the command line, the same model can be selected and designed by typing:

```
> sepsis.dsmb2 <- update(sepsis.dsmb,
+   early.stopping="both")
```

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.3, 0.3), alt.hypothesis = "lower", nbr.analyses
= 4, sample.size = 1700, test.type = "less", size =
0.0225, power = 0.9, early.stopping = "both", P = 1)

PROBABILITY MODEL and HYPOTHESES:
Two arm study of binary response variable
Theta is difference in probabilities (Treatment - Comparison)
One-sided hypothesis test of a lesser alternative:
Null hypothesis : Theta >= 0          (size = 0.0225 )
Alternative hypothesis: Theta<=-0.07146 (power= 0.9 )
(Emerson & Fleming (1989) symmetric test)
```

```
STOPPING BOUNDARIES: Sample Mean scale
                     a      d
Time 1 (N= 425) -0.1745  0.0872
Time 2 (N= 850) -0.0872  0.0000
Time 3 (N= 1275) -0.0582 -0.0291
Time 4 (N= 1700) -0.0436 -0.0436
```

From the command line, you can display the same information about `sepsis.dsmb2` using the `print` function:

```
> print(sepsis.dsmb2)
```

Evaluating the Design Compare the two designs considered by the DSMB.

1. Select the **Plot** tab and choose the **Decision Boundaries** plot type.
2. In the **Designs To Plot** group box, click the **Refresh** and **Deselect All** buttons, then select `sepsis.dsmb` to compare with the current design (`sepsis.dsmb2`).
3. In the **Options** groupbox, choose the **Overlay Designs** option.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(sepsis.dsmb2, sepsis.dsmb,
+     superpose.design=T)
```

The result is shown in Figure 12.9.

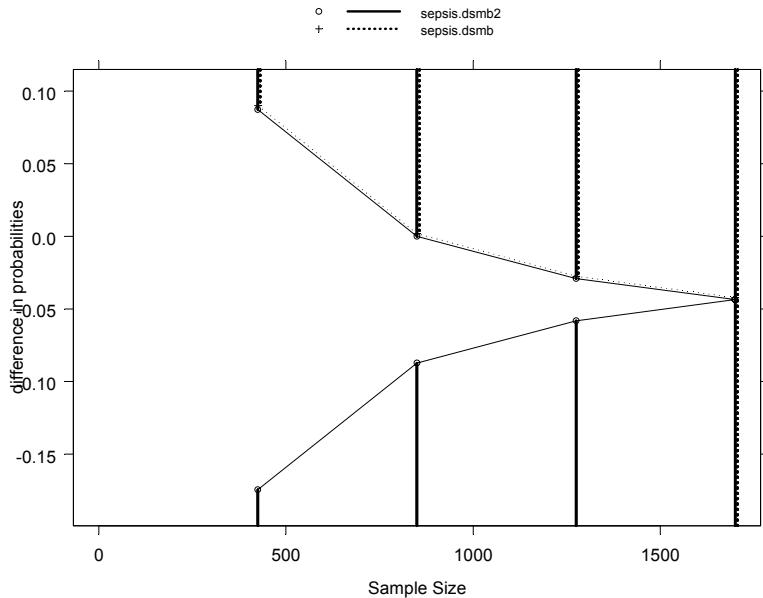


Figure 12.9: A comparison of stopping rules for designs `sepsis.dsmb2` and `sepsis.dsmb`.

The upper boundaries for the two designs are virtually indistinguishable. Design `sepsis.dsmb2` includes a symmetric lower boundary to stop for efficacy.

Now compare the power curves.

1. Deselect the **Decision Boundaries** plot type.
2. Select the **Power Curve** plot type.
3. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotPower(sepsis.dsmb2, sepsis.dsmb)
```

The result is shown in Figure 12.10.

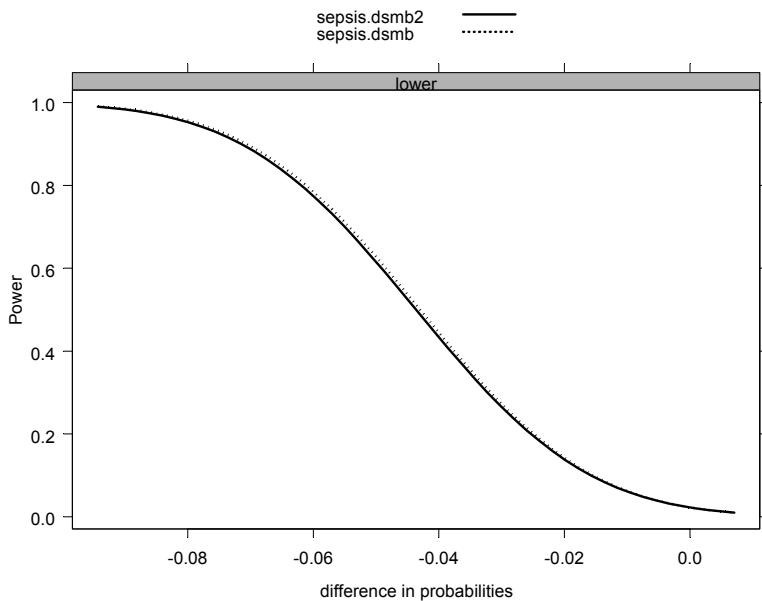


Figure 12.10: The power curves for designs `sepsis.dsmb2` and `sepsis.dsmb`.

The curves are virtually identical. To get better resolution, compare the power curves by plotting the difference between them.

1. Deselect the **Power Curve** plot type.
2. Select the **Power Curve vs. Reference** plot type.
3. Set the **Reference Design** to `sepsis.dsmb`.

Click **Apply**. The result is shown in Figure 12.11.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(sepsis.dsmb2, sepsis.dsmb,
+   reference="sepsis.dsmb")
```

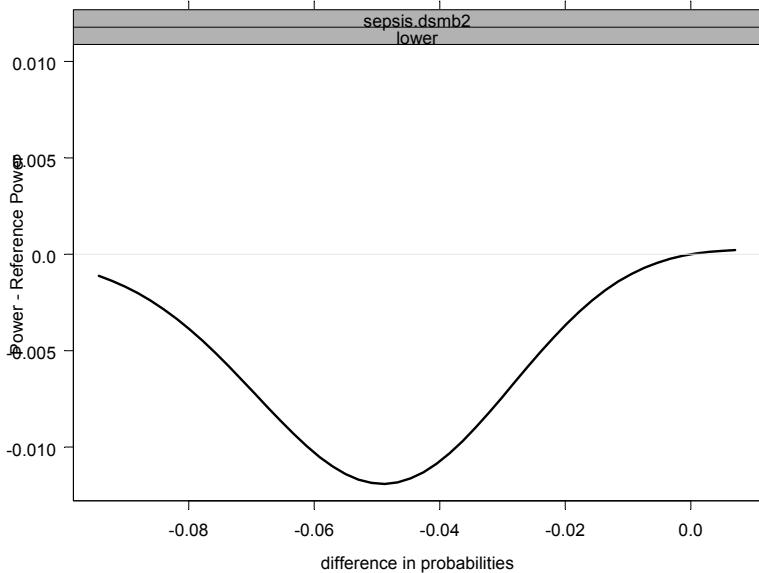


Figure 12.11: The difference curve created by subtracting design `sepsis.dsmb` from design `sepsis.dsmb2`.

The loss of power is rarely more than 1%.

Now compare the sample size distributions for these designs.

1. Deselect the **Power Curve vs. Reference** plot type.
2. Select the **Average Sample Number** plot type.
3. In the **Designs To Plot** group box, select `sepsis.fix` in addition to `sepsis.dsmb`.
4. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotASN(sepsis.dsmb2, sepsis.dsmb, sepsis.fix,
+           fixed=F)
```

The result is shown in Figure 12.12.

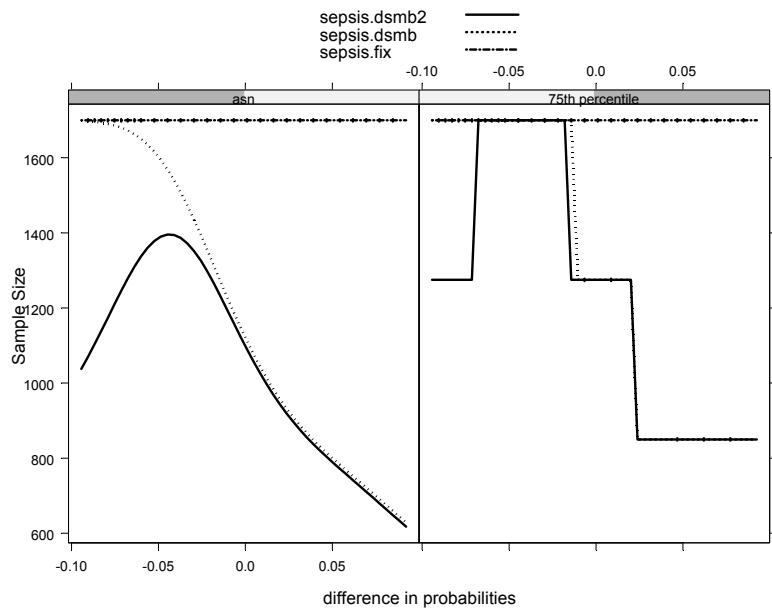


Figure 12.12: The average sample size (ASN) curve and the 75th %-tile curve for sequential designs `sepsis.dsmb2` and `sepsis.dsmb`, and fixed sample design `sepsis.fix`.

Design `sepsis.dsmb2` has a significantly lower average sample size when the treatment is significantly effective, with only a negligible loss in power. Therefore, the DSMB tentatively adopted this design.

SPONSOR FUTILITY BOUNDARIES

When the sponsors were shown the potential gains in efficiency with negligible loss of power associated with the one-sided symmetric design, they questioned whether additional efficiency could be gained by moving the upper (futility) boundary even lower. They therefore explored three asymmetric designs.

Specifying the Designs

You can create these three designs by adjusting the boundary shape parameter P. The parameter P determines the tradeoff between early conservatism and the efficiency of the design. Larger values of P correspond to greater early conservatism.

Let's try adjusting the value of P for the upper ("d") boundary, gradually decreasing it 0.9 to 0.75 to 0.5.

1. Deselect the **Average Sample Number** plot type.
2. Select the **Design** tab, and **Null Bnd. Shape** to **P = .9**
3. Save the first design object under the name **sepsis.futility** using the **Save As** field.

From the command line, the same model can be designed and printed by typing:

```
> sepsis.futility <- update(sepsis.dsmb2, P=c(1,.9))
> print(sepsis.futility)
```

Click **Apply** to create the design object and print out summary information in a report window:

```
Call:
seqDesign(prob.model = "proportions", arms = 2, null.hypothesis
= c(0.3, 0.3), alt.hypothesis = "lower", nbr.analyses
= 4, sample.size = 1700, test.type = "less", size =
0.0225, power = 0.9, early.stopping = "both", P = c(1,
0.9))
```

PROBABILITY MODEL and HYPOTHESES:

Two arm study of binary response variable

Theta is difference in probabilities (Treatment - Comparison)

One-sided hypothesis test of a lesser alternative:

```
Null hypothesis : Theta >= 0          (size = 0.0225 )
Alternative hypothesis : Theta <= -0.07172 (power = 0.9 )
```

```
STOPPING BOUNDARIES: Sample Mean scale
                     a          d
Time 1 (N= 425) -0.1739  0.0662
Time 2 (N= 850) -0.0869 -0.0052
Time 3 (N= 1275) -0.0580 -0.0304
Time 4 (N= 1700) -0.0435 -0.0435
```

Now lower the value of P even further. This time we want to set the P to .75, which is not an option on the **Design** tab, so we'll use the **Advanced** tab.

1. Select the **Advanced** tab, and set the **P** field in the **Shape Parameters** groupbox equal to **1, Inf, Inf, 0.75**.
2. Select the **Design** tab, and save the second design object under the name **sepsis.futility2** using the **Save As** field.
3. Click **Apply**.

Finally, lower the value of P all the way to .5.

1. Select the **Design** tab, and **Null Bnd. Shape** to **Pocock (P = .5)**.
2. Save the third design object under the name **sepsis.futility3** using the **Save As** field.
3. Click **Apply**.

From the command line, type:

```
> sepsis.futility2 <- update(sepsis.dsmb2, P=c(1,.75))
> sepsis.futility3 <- update(sepsis.dsmb2, P=c(1,.5))
```

Evaluating the Designs

Let's compare the stopping boundaries for the three new designs with the boundaries for the design proposed by the DSBM.

1. Select the **Plot** tab.
2. Select the **Decision Boundaries** plot type.
3. In the **Designs To Plot** group box, click the **Refresh** and **Deselect All** buttons, then select the comparison design objects **sepsis**, **futility2**, **sepsis.futility**, and **sepsis.dsmb2**.

4. In the **Options** groupbox, ensure that the **Overlay Designs** option is selected.
5. Click **Apply**.

From the command line, the same plot can be created by typing:

```
> seqPlotBoundary(sepsis.dsmb2, sepsis.futility,
+     sepsis.futility2, sepsis.futility3,
+     superpose.design=T)
```

The result is shown in Figure 12.13.

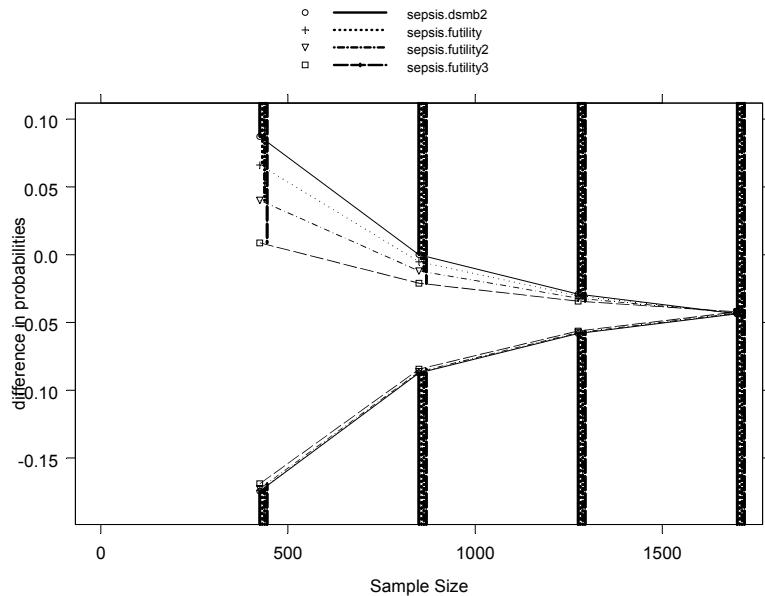


Figure 12.13: A comparison of stopping rules for designs with increasingly less conservative upper stopping boundaries.

Notice how the upper boundary becomes less conservative as the shape parameter P is decreased from 1 to .9 to 0.75 to .5.

Now compare the ASN and power curves.

1. Deselect the **Decision Boundaries** plot type.

2. Select the **Power Curve** and **Average Sample Number** plot types.
3. Click **Apply**.

From the command line, the same plots can be created by typing:

```
> seqPlotASN(sepsis.dsmb2, sepsis.futility,
+           sepsis.futility2, sepsis.futility3, fixed=F)
> seqPlotPower(sepsis.dsmb2, sepsis.futility,
+             sepsis.futility2, sepsis.futility3)
```

The results are shown in Figure 12.14 and Figure 12.15.

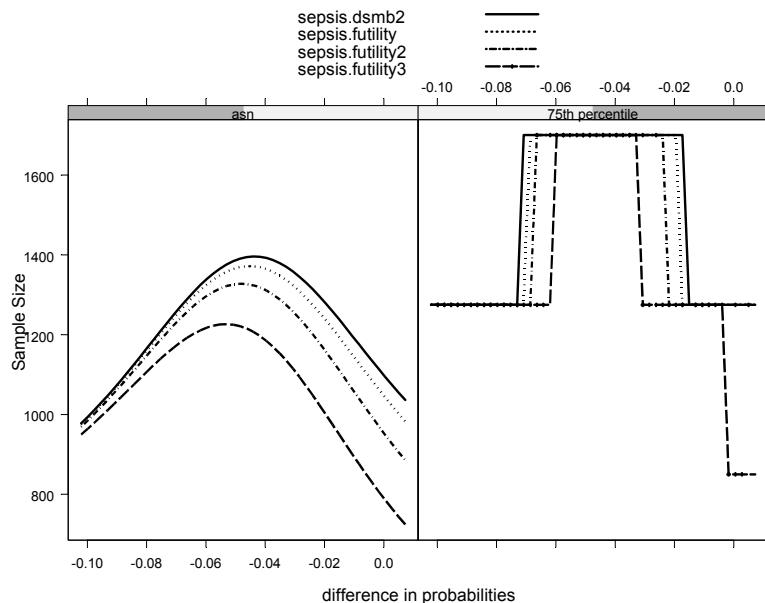


Figure 12.14: The average sample size (ASN) curve and the 75th %-tile curve for designs with increasingly less conservative upper stopping boundaries.

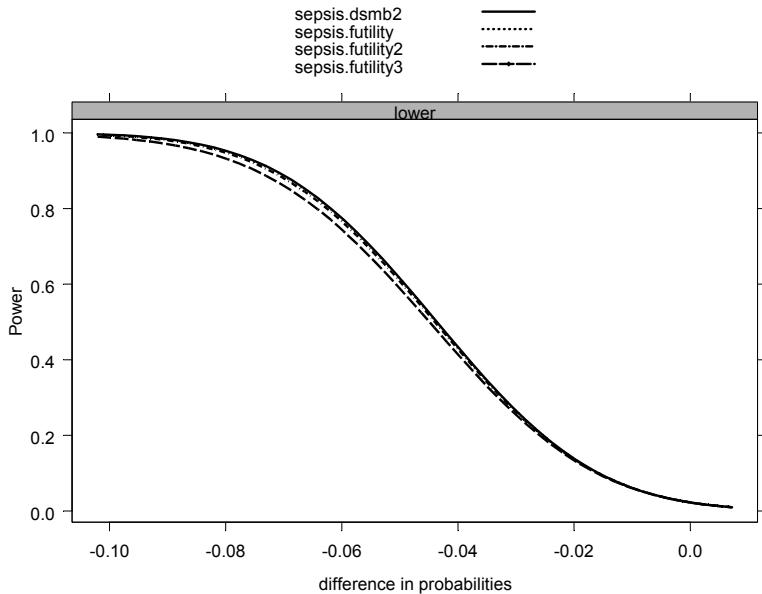


Figure 12.15: The power curves for designs with increasingly less conservative upper stopping boundaries.

The ASN curves show better efficiency—lower average sample size when the treatment is beneficial—as the shape parameter P is decreased from 1 to .9 to 0.75 to .5.

The power curves are difficult to distinguish. To get better resolution, compare the power curves for the three new designs by plotting the difference between them and the design proposed by the DSMB.

1. Deselect the **Power Curve** and **Average Sample Number** plot types.
2. Select the **Power Curve vs. Reference** plot type.
3. Set the **Reference Design** to `sepsis.dsmb2`.

Click **Apply**. The result is shown in Figure 12.16.

From the command line, you can create the same plot by typing:

```
> seqPlotPower(sepsis.futility, sepsis.futility2,
+               sepsis.futility3, sepsis.dsmb2,
+               reference="sepsis.dsmb2")
```

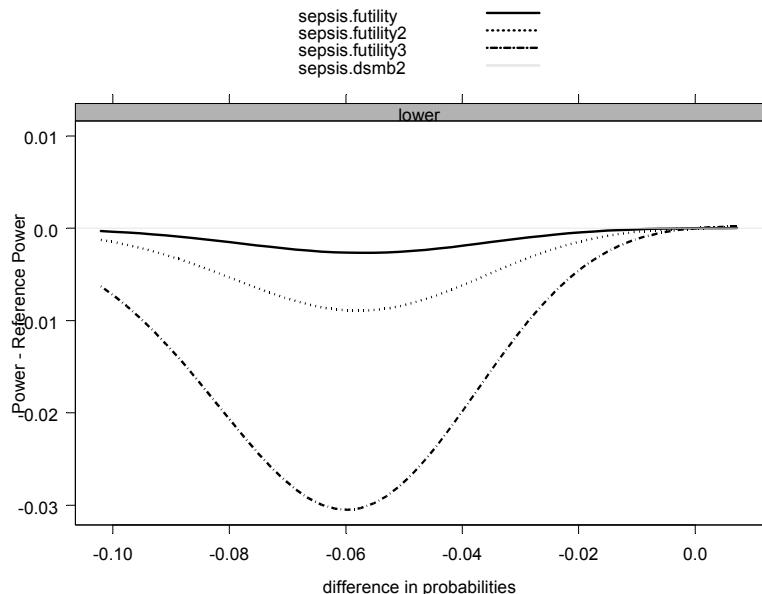


Figure 12.16: The difference curves created by subtracting design `sepsis.dsmb2` from designs with increasingly less conservative upper stopping boundaries.

The loss of power is less than 1% for $P=.75$, but increases rapidly as P is decreased further. Based on these considerations, the sponsors selected the asymmetric design `sepsis.futility2`, where $P=.75$. This design offered the best trade-off of efficiency and power.

ADVANCED BOUNDARY SCALES

To further confirm their choice, the sponsors also looked at the stopping boundaries for design `sepsis.futility2` on other scales.

Conditional Futility Scale

First, they wanted to evaluate the stopping rule with respect to the probability that the decision made when stopping at some interim analysis might be different than the decision which would have been reached had the study not been terminated prematurely. Evaluations of the stopping rule with respect to these criteria are based on the distribution of the test statistic at the final analysis conditional upon the test statistic being equal to the stopping boundary at an interim analysis.

For example, because each stopping boundary is associated with rejection of a particular hypothesis, the sponsors wanted to consider the conditional probabilities under the corresponding hypothesis. These boundaries can be obtained using the **Conditional Futility** scale.

1. Choose **SeqTrial ▶ Update Design**.
2. Select `sepsis.futility2` from the **Select Design** listbox, then click **OK**.
3. Select the **Results** tab.
4. Set the **Display Scale** to **Conditional Futility**.
5. Click **Apply**.

From the command line, the same results can be printed by typing:

```
> changeSeqScale(sepsis.futility2, "C")
```

Chapter 12 Case Study #4: Efficiency Concerns

STOPPING BOUNDARIES: Conditional Probability scale
(Conditional probability that estimated treatment effect at the last analysis would correspond to an opposite decision computed using hypothesized true treatment equal to hypotheses being rejected)

a	d
Time 1 (N= 425)	0.5 0.7651
Time 2 (N= 850)	0.5 0.6847
Time 3 (N= 1275)	0.5 0.6166
Time 4 (N= 1700)	0.5 0.5000

The “a” boundary indicates the efficacy of the new treatment. In this case, if the null hypothesis were true (that is, the treatment were no better than placebo), and you were to observe an effect right on the “a” boundary (indicating that you should stop and declare benefit), you would stand a 50% chance of reversing that decision at the final analysis.

Similarly, the “d” boundary indicates the futility of showing that the new treatment is beneficial. In this case, if the alternative hypothesis were true (that is, the treatment were more than 7% better than placebo), and you were to observe an effect right on the “d” boundary at the first interim analysis (indicating that you should stop and declare insufficient benefit), you would stand a 77% chance of reversing that decision at the final analysis. At the second and third interim analyses, you would stand a 68% and 62% chance of reversing your decision at the final analysis, respectively.

Obviously, these probabilities are very high. This is because if the null hypothesis were true, you’d be quite unlikely to stop on the “a” boundary, and if the alternative hypothesis were true, you’d be quite unlikely to stop on the “d” boundary. Hence, computing the conditional probabilities by assuming the opposite hypothesis is likely to be too pessimistic at the “a” boundary, and too optimistic at the “d” boundary. A better approach is to use your best estimate of the treatment effect at the place you’re actually stopping.

1. Enter **hypTheta="estimate"** in the **Scale Parameters** field under the **Display Scale**.
2. Click **Apply**.

From the command line, the same results can be printed by typing:

```
> changeSeqScale(sepsis.futility2, seqScale("C",
+      hypTheta="estimate"))
```

STOPPING BOUNDARIES: Conditional Probability scale
 (Conditional probability that estimated treatment effect at the last analysis would correspond to an opposite decision computed using hypothesized true treatment equal to maximum likelihood estimate)

a	d
---	---

Time 1 (N= 425) 0.0000 0.0000
 Time 2 (N= 850) 0.0021 0.0196
 Time 3 (N= 1275) 0.0884 0.1516
 Time 4 (N= 1700) 0.5000 0.5000

Now the probabilities for reversing a decision are quite low.

Predictive Futility Scale

On the **Predictive Futility** scale, you can also evaluate the stopping boundaries with respect to the predictive probability that the opposite decision might be made at the final analysis, where the predictive probability is computed by conditioning on the value of the test statistic at the boundary and averaging over the posterior distribution based on a prior distribution for μ . Lacking any information to the contrary, you could choose a noninformative prior.

1. Set the **Display Scale** to **Predictive Futility**.
2. Clear the **Scale Parameters** field, then enter **priorTheta=0, priorVariation=Inf**. Setting the **priorVariation** argument to Inf indicates a noninformative prior.
3. Click **Apply**.

From the command line, the same results can be printed by typing:

```
> changeSeqScale(sepsis.futility2, seqScale("H",
+      priorTheta=0, priorVariation=Inf))
```

STOPPING BOUNDARIES: Predictive Probability scale
 (Predictive probability that estimated treatment effect at the last analysis would correspond to an opposite decision based on prior distribution having median 0 and variation parameter Inf)

a	d
---	---

```
Time 1 (N= 425) 0.0002 0.0120
Time 2 (N= 850) 0.0214 0.0725
Time 3 (N= 1275) 0.1211 0.1863
Time 4 (N= 1700) 0.5000 0.5000
```

Bayesian Scale

The Bayesian properties of a particular stopping rule can be evaluated for a specified prior by considering the posterior probabilities of the various hypotheses. The Bayesian scale allows you to assess whether a decision made by a frequentist stopping rule is compatible with an inference that considers the prior probability that the treatment might be beneficial. Frequentist decisions are based solely on whether the observed data are commonly observed under specific hypotheses. Bayesian inference considers the more interesting question of whether a specific hypothesis is very probable given the observed data. By examining these Bayesian posterior probabilities under a variety of priors, you can assess whether a rejection of the null hypothesis is more likely to be a Type I error or a correct identification of the alternative hypothesis.

1. Set the **Display Scale** to **Bayesian**.
2. Change the **Scale Parameters** field to read **priorTheta=-.04**, **priorVariation=.1**.
3. Click **Apply**.

From the command line, the same results can be printed by typing:

```
> changeSeqScale(sepsis.futility2, seqScale("B",
+     priorTheta=-.04, priorVariation=.1))
```

```
STOPPING BOUNDARIES: Bayesian scale
(Posterior probability of hypotheses based on prior
distribution having median -0.04
and variation parameter 0.1)
      a      d
Time 1 (N= 425) 1.0000 0.9987
Time 2 (N= 850) 0.9979 0.9945
Time 3 (N= 1275) 0.9904 0.9893
Time 4 (N= 1700) 0.9788 0.9839
```

APPENDIX A: FUNCTIONS BY CATEGORY

This appendix gives a list of functions for S+SEQTRIAL by category. Documentation for these functions is available both in the function reference in this manual (see Appendix B) and on-line using the `help` function.

Appendix A: Functions by Category

Computing Group Sequential Designs

as.seqBoundary	Create a Boundary Object
as.seqScale	Create a Scale Object
seqBoundary	Create a Boundary Object
seqDesign	Group Sequential Design
seqFixDesign	Convert a Sequential Design to a Fixed Design
seqScale	Create a Scale Object
update.seqDesign	Update a Group Sequential Design

Details about calling seqDesign

seqDesign.boundaryShape	Arguments: design.family, P, R, A, G, early.stopping, exact.constraint, minimum.constraint, maximum.constraint
seqDesign.probabilityModel	Arguments: prob.model, arms, log.transform, null.hypothesis, alt.hypothesis, variance, ratio
seqDesign.testSpecifications	Arguments: test.type, size, conf.level, power, alpha, beta, epsilon
seqDesign.specialFamilies	Designs in the group sequential literature.
seqDesign.multiple.arms	Multiple-arm trials (more than 2 arms)

Evaluating Designs

plot.seqDesign	Plot Group Sequential Design Boundaries
plot.seqOperatingChar	Plot Any of the Operating Characteristics
plot.summary.seqDesign	Plot Boundaries and Operating Characteristics
seqInference	Inference for a Hypothesis Test
seqOperatingChar	Operating Characteristics (power, ASN, stopping probabilities)
seqPHSubjects	Convert #Events to #Subjects for Proportional Hazards Model
seqPlotASN	Plot ASN and Sample Size Quantile Curves
seqPlotBoundary	Plot Group Sequential Design Boundaries
seqPlotPower	Plot a Power Curve
seqPlotStopProb	Plot Stopping Probabilities
seqSampleSizeQuantile	Quantiles of Sample Size
summary.seqDesign	Summary of Group Sequential Design
summary.seqPHSubjects	Details of #Events to #Subjects conversion, including interim analyses.

Monitoring a Sequential Trial

plot.seqMonitor
seqMonitor

Plotting for Monitoring a Trial
Monitor a Group Sequential Trial

High-Level Plot Functions

plot.seqBoundary
plot.seqDesign
plot.seqInference
plot.seqMonitor
plot.seqOperatingChar

seqPlotASN
seqPlotBoundary
seqPlotInference
seqPlotPower
seqPlotStopProb

Plot Boundaries for a Group Sequential Design
Plot Boundaries for a Group Sequential Design
Plot Inferences for a Group Sequential Design
Plot Boundaries and Observed Values
Plot Operating Characteristics for a Group
Sequential Design
Plot ASN and Sample Size Quantile Curves
Plot Boundaries for a Group Sequential Design
Plot a Group Sequential Design
Plot a Power Curve
Plot Stopping Probabilities

Statistical Inference

confidenceInterval.seqDesign

plot.seqInference
seqInference
seqPlotInference
summary.seqDesign

Confidence interval for a Group Sequential
Design
Plot inferences for a Group Sequential Design
Inference for a Hypothesis Test
Plot inferences for a Group Sequential Design
Summary of Group Sequential Design

Distribution of a Test Statistic

dSeq
pSeq

qSeq
meanSeq

Density of a Test Statistic
Cumulative Distribution Function of a Test
Statistic
Quantiles of a Test Statistic
Mean of a Test Statistic

Extracting Items from Group Sequential Design Objects

confidenceLevel
sampleSize
seqBoundary

seqExtract

The Confidence Level for a Hypothesis Test
Sample Size
Extract a Boundary Object from a Group
Sequential Design
Extract various components

Appendix A: Functions by Category

Data Manipulation

seqDF.ASN	Create a Data Frame for Plotting
seqDF.seqBoundary	Create a Data Frame for Plotting
seqDF.seqInference	Create a Data Frame for Plotting
seqDF.Power	Create a Data Frame for Plotting
seqDF.seqDesign	Create a Data Frame for Plotting
seqDF.StopProb	Create a Data Frame for Plotting

Methods and Generic Functions for Group Sequential Analysis

confidenceInterval	Confidence Interval
confidenceLevel	The Confidence Level for a Hypothesis Test
plot	Plot an object
powerTest	The Power for a Hypothesis Test
print	Print an Object
sampleSize	Sample Size
summary	Summary an Object
update	Update a Model or Design

Objects and Data Types

seqBoundary.object	Decision Boundary Object
seqDesign.object	Group Sequential Design Object
seqHypothesis.object	Hypotheses Object
seqInference.object	Inference Object
seqModel.object	Sequential Design Model Object
seqOperatingChar.object	Operating Characteristics Object
summary.seqDesign.object	Group Sequential Design Summary Object

Utilities

all.equal.seqDesign	Compare two seqDesign objects
SeqTrial.options	Set Options for seqtrial Library

Miscellaneous Low Level Functions

checkSeqBoundary	Check Validity of a Boundary Object
checkSeqScale	Checks if Display Scale is Valid
getListNames	Utility Function, Handles Optional Arguments
ifelse1	Conditional Data Selection
NaTo0	Replace Missing Values with Zero.
pasteParens	Paste Arguments Inside Parentheses

redundantList	Determine if Elements of a List are Redundant
sampleSize.quantile.curve	Sample Size Quantile Curve
seqBoundaryLabels	Label for the Boundary Scale
seqBoundaryTimeValue	Convert a seqBoundary Object to Time-Value Pairs
seqDesignCtoS	Wrapper for Interface to C Code
seqGetKeyParameters	Get Trellis Parameters for Key
seqParseDotargs	Utility Function, Handles Optional Arguments

Low-Level Plot Functions

panel.seqBoundary	Panel Function for Decision Boundaries
panel.seqBoundary.superpose	Panel Function for Superposing Boundaries
panel.seqInference	Panel Function for Inferences
panel.seqPlotPower	Panel Function for Power Curves
panel.seqPlotStopProb	Panel Function for Stopping Probabilities

Command-line code from this manual

chapter1.seqtrial	Command-line code for chapter 1
chapter2.seqtrial	Command-line code for chapter 2
chapter3.seqtrial	Command-line code for chapter 3
chapter4.seqtrial	Command-line code for chapter 4
chapter5.seqtrial	Command-line code for chapter 5
chapter6.seqtrial	Command-line code for chapter 6
chapter7.seqtrial	Command-line code for chapter 7
chapter8.seqtrial	Command-line code for chapter 8
chapter9.seqtrial	Command-line code for chapter 9
chapter10.seqtrial	Command-line code for chapter 10
chapter11.seqtrial	Command-line code for chapter 11
chapter12.seqtrial	Command-line code for chapter 12
tutorial.seqtrial	Same as chapter3.seqtrial
case1.seqtrial	Same as chapter9.seqtrial
case2.seqtrial	Same as chapter10.seqtrial
case3.seqtrial	Same as chapter11.seqtrial
case4.seqtrial	Same as chapter12.seqtrial

Appendix A: Functions by Category

APPENDIX B: FUNCTION REFERENCE

This section describes the functions included with S+SEQTRIAL.

The information in this section duplicates that found in the online Help. For instructions on how to access the online Help, see page 20.

all.equal.seqDesign Test Two seqDesign Objects for Full Equality

DESCRIPTION

Returns TRUE or a message about what is different between the two objects.

```
all.equal.seqDesign(target, current, ..., compareCall = T,  
                    compareDate = F, compareDialog = F)
```

REQUIRED ARGUMENTS

target, current: two "seqDesign" or "seqMonitor" objects, to be tested for equivalence.

OPTIONAL ARGUMENTS

...: other arguments to **all.equal**, e.g. **tolerance** for comparing numerical components.

compareCall: if FALSE, then the **call** and **specification** components are ignored when checking for differences. This is useful for comparing two objects which are equivalent except for trivial differences in how **seqDesign** was called when creating the objects. Similarly, the **seqDesignCall** component of "seqMonitor" objects are ignored.

compareDate: if FALSE then differences in **dialog.info** components (created by the graphical interface for SeqTrial) are ignored when comparing objects.

compareDialog: if FALSE then differences in **dialog.info** components (created by the graphical interface for SeqTrial) are ignored when comparing objects.

VALUE

If all components and attributes of the two objects test equal, returns TRUE . Otherwise, returns a character vector giving all the discovered differences (excluding ignored components).

DETAILS

This sets certain components of the objects to NULL, then calls other methods for **all.equal**.

SEE ALSO

all.equal (located in S-PLUS Help), **seqDesign.object** .

EXAMPLE

```
des <- seqDesign(alt.hypothesis = 0.1, nbr.analyses = 3)  
des2 <- update(des, test.type = "greater") # superfluous argument  
all.equal(des, des2, compareCall=F) # no non-trivial differences  
all.equal(des, des2)  
all.equal(des, des2, compareDate=T) # all differences
```

case1.seqtrial Commands for Running Case Study #1

DESCRIPTION

The script shown below contains the command-line functions for running Case Study #1: Hybrid Tests, as described in the *S+SeqTrial User's Guide*.

EXAMPLE

```
#LIMITATIONS OF THE STANDARD DESIGNS
ssr.ofb <- seqDesign(prob.model = "proportions", arms = 2, size = 0.05,
                      power = 0.95, sample.size = 200, null.hypothesis = c(0.5, 0.5),
                      alt.hypothesis="calculate", variance = "null", test.type = "two.sided",
                      nbr.analyses = 4, P = 1, display.scale = "S")
print(ssr.ofb)

ssr.poc <- update(ssr.ofb, P=c(1.0,Inf,Inf,0.5))
print(ssr.poc)

ssr.p0.1 <- update(ssr.ofb, P=c(1.0,Inf,Inf,0.1))
print(ssr.p0.1)

seqPlotBoundary(ssr.p0.1, ssr.ofb, ssr.poc, display.scale="S")
seqPlotASN(ssr.p0.1, ssr.ofb, ssr.poc, fixed=F)
seqPlotPower(ssr.p0.1, ssr.ofb, ssr.poc)
seqPlotStopProb(ssr.p0.1, ssr.ofb, ssr.poc)

#HYBRID TESTS
ssr.eps.8 <- update(ssr.poc, epsilon= c(1, 0.8))
print(ssr.eps.8)

ssr.eps.5 <- update(ssr.poc, test.type="advanced", epsilon=c(1, 0.5))
print(ssr.eps.5)

ssr.eps.2 <- update(ssr.poc, test.type="advanced", epsilon=c(1, 0.2))
print(ssr.eps.2)

ssr.oneside <- update(ssr.poc, test.type="advanced", epsilon= c(1, 0))
print(ssr.oneside)

seqPlotBoundary(ssr.oneside, ssr.ofb, ssr.poc, ssr.eps.8,
                ssr.eps.5, ssr.eps.2)
seqPlotASN(ssr.oneside, ssr.ofb, ssr.poc, ssr.eps.8, ssr.eps.5,
            ssr.eps.2, fixed=F)
seqPlotPower(ssr.oneside, ssr.ofb, ssr.poc, ssr.eps.8,
            ssr.eps.5, ssr.eps.2)
```

case2.seqtrial Commands for Running Case Study #2

DESCRIPTION

The script shown below contains the command-line functions for running Case Study #2: Unplanned Analyses, as described in the *S+SeqTrial User's Guide*.

REFERENCES

Wang, S. K. and Tsiatis, A. A. (1987) Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics*, Vol 43, 193-199.

EXAMPLE

```
#DESIGNING A FIXED SAMPLE TEST
aml.fix <- seqDesign(prob.model = "proportions", arms = 2, size = 0.05,
                      power = 0.80, null.hypothesis = c(0.60, 0.60),
                      alt.hypothesis = c(0.80, 0.60), variance = "alternative",
                      test.type = "two.sided")
print(aml.fix)

aml.fix <- seqDesign(prob.model = "proportions", arms = 2, size = 0.05,
                      power = 0.80, null.hypothesis = c(0.60, 0.60),
                      alt.hypothesis = "calculate", variance = "alternative",
                      test.type = "two.sided", sample.size=180)
print(aml.fix)

#DESIGNING A GROUP SEQUENTIAL TEST
aml.obf <- update(aml.fix, nbr.analyses=3, early.stopping="alternative",
                   sample.size=c(90,130,180))
print(aml.obf)

seqPlotBoundary(aml.obf, fixed=F)
seqPlotPower(aml.obf,aml.fix, superpose.design=T)
seqPlotPower(aml.obf, aml.fix, reference="aml.fix")

#CONDUCTING THE CLINICAL TRIAL
cr.idr <- c(rep(1,35), rep(0,10))
cr.dnr <- c(rep(1,25), rep(0,20))
cr1 <- c(cr.idr, cr.dnr)
tx1 <- c(rep(1,45), rep(0, 45))

first.analysis <- seqMonitor(aml.obf, cr1, tx1)
print(first.analysis)
plot(first.analysis)

first.analysis <- seqMonitor(aml.obf, cr1, tx1, future.analyses=c(130,180))
print(first.analysis)
plot(first.analysis)
```

```

seqPlotPower(first.analysis,aml.obf,reference="aml.obf")

cr.idr <- c(rep(1,16), rep(0,4))
cr.dnr <- c(rep(1,13), rep(0,7))
cr2 <- c(cr1, cr.idr, cr.dnr)
tx2 <- c(tx1, rep(1,20), rep(0,20))
second.analysis <- seqMonitor(first.analysis, cr2, tx2, future.analyses=180)
print(second.analysis)
plot(second.analysis)

aml.fix130 <- update(aml.fix, sample.size=130)
seqMonitor(aml.fix130, cr2, tx2)

#PERFORMING A SENSITIVITY ANALYSIS
shift.boundary <- function(design, k, value)

  ## This function shifts the upper boundaries to be
  ## equal to value at the kth analysis. This function
  ## only works for the special case of Wang & Tsiatis
  ## design. Value is on the sample mean scale.
  P <- design$parameters$P[1]
  N <- design$parameters$sample.size
  PI <- N / N[length(N)]
  G <- value * PI[k] ^ P
  bnd <- seqBoundary(design, display.scale="X")
  bnd[,4] <- PI ^ (-P) * G
  bnd[,1] <- - bnd[,4]
  bnd[,2:3] <- 0
  bnd[length(N),2] <- -G
  bnd[length(N),3] <- G
  update(design, exact.constraint=bnd)

aml.best <- shift.boundary(aml.obf, 2, .2)
print(aml.best)

aml.worst <- shift.boundary(aml.obf, 1, 0.2223)
print(aml.worst)

seqPlotBoundary(aml.obf, aml.best, aml.worst, observed=c(10/45, 14/65),
  N=c(90,130), fixed=F)

seqInference(aml.best, 2, .2, "X")
seqInference(aml.worst, 2, .2, "X")

aml.obf1 <- update(aml.obf, nbr.analyses=4, sample.size= c(50,90,130,180))
aml.best1 <- shift.boundary(aml.obf1, 3, .2)
print(aml.best1)
seqInference(aml.best1, 3, .2, "X")

```

```
aml.worst1 <- shift.boundary(aml.obf1, 2, .2223)
print(aml.worst1)
seqInference(aml.worst1, 3, .2, "X")

aml.poc <- update(aml.obf, P=0.5)
aml.best2 <- shift.boundary(aml.poc, 2, .2)
print(aml.best2)
seqInference(aml.best2, 2, .2, "X")

aml.worst2 <- shift.boundary(aml.poc, 1, .2223)
print(aml.worst2)
seqInference(aml.worst2, 2, .2, "X")
```

case3.seqtrial Commands for Running Case Study #3

DESCRIPTION

The script shown below contains the command-line functions for running Case Study #3: Unexpected Toxicities, as described in the *S+SeqTrial User's Guide*.

EXAMPLE

```
#DESIGNING A FIXED SAMPLE TEST
toxic.fix <- seqDesign(prob.model = "proportions", arms = 1,
                        null.hypothesis = 0.2, alt.hypothesis="calculate",
                        variance = "alternative", test.type = "greater", sample.size = 35,
                        size = 0.025, power = 0.975)
print(toxic.fix)

#DESIGNING GROUP SEQUENTIAL TESTS
toxic.obf <- update(toxic.fix, nbr.analyses=7, early.stopping="alternative",
                      P=1)
print(toxic.obf)

toxic.obf <- update(toxic.obf, display.scale="S")
print(toxic.obf)

toxic.poc <- update(toxic.obf, P=0.5)
print(toxic.poc)

seqPlotBoundary(toxic.poc, toxic.obf, superpose.design=T, display.scale="S")
seqPlotStopProb(toxic.poc, toxic.obf)
seqPlotASN(toxic.poc, toxic.obf, fixed=F)

summary(toxic.poc)
summary(toxic.obf)

#DATA-IMPOSED STOPPING RULES
cond.size <- function(des, theta)

  if(length(theta) != 1) stop("theta must be scalar")
  bnd <- seqBoundary(des, scale="S")
  N <- sampleSize(des)-4
  ## compute P(reject H(0) | Y=y, C=c)
  ## rows correspond to y = 0:4
  ## columns correspond to C = 0:5
  cond.size <- matrix(0, nrow=5, ncol=6)
  for(y in 0:3)
    cond.size[y+1, (y+2):6] <- 1-pbinom(12-y, size=31,
                                           prob=theta)
  new <- update(des, exact.constraint=bnd-y, sample.size=N)
  cond.size[y+1, 1:(y+1)] <- seqOperatingChar(new,
```

```

theta)[,"power.upper"]

cond.size[5,1:5] <- 1
cond.size[5,6] <- 1-pbinom(8, size=31,
                           prob=theta)
## multiply by P(Y=y) and
## sum over y to get conditional size (power)
dimnames(cond.size) <- list(rep("",5), paste("C=",0:5,sep=""))
colSums(cond.size * dbinom(0:4, size=4, prob=theta))

round(cond.size(toxic.poc, .2),4)
round(cond.size(toxic.poc, .5),4)

```

case4.seqtrial Commands for Running Case Study #4

DESCRIPTION

The script shown below contains the command-line functions for running Case Study #4: Efficiency Concerns, as described in the *S+SeqTrial User's Guide*.

EXAMPLE

```
#INITIAL PROTOCOL
sepsis.fix <- seqDesign(prob.model="proportions", arms=2, size=.045, power=.9,
    null.hypothesis= c(.30, .30), alt.hypothesis=c(.23, .30),
    test.type="two.sided")
print(sepsis.fix)

sepsis.fix <- update(sepsis.fix, sample.size=1700, alt.hypothesis="lower")
print(sepsis.fix)

#SPONSOR SAFETY BOUNDARY
bound <- seqBoundary(scale = "Z", rbind(c(-Inf, 0, 0, 1.6450),
    c(-2.0047,-2.0047,2.0047,2.0047)))
sepsis.safety <- update(sepsis.fix, nbr.analyses=2, exact.constraint=bound)
print(sepsis.safety)

seqPlotASN(sepsis.safety, sepsis.fix, superpose.design=T, fixed=F)
seqPlotPower(sepsis.safety, sepsis.fix)
seqPlotPower(sepsis.safety, reference=sepsis.fix)

#DSMB SAFETY BOUNDARY
sepsis.dsmb <- update(sepsis.safety, size=0.0225, test.type="less",
    nbr.analyses=4, P=1, early.stopping="null", exact.constraint=)
print(sepsis.dsmb)

seqPlotBoundary(sepsis.dsmb, sepsis.safety, superpose.design=T)
seqPlotPower(sepsis.dsmb, sepsis.safety)
seqPlotASN(sepsis.dsmb, sepsis.safety, sepsis.fix, fixed=F)

#DSMB EFFICIENCY BOUNDARY
sepsis.dsmb2 <- update(sepsis.dsmb, early.stopping="both")
print(sepsis.dsmb2)

seqPlotBoundary(sepsis.dsmb2, sepsis.dsmb, superpose.design=T)
seqPlotPower(sepsis.dsmb2, sepsis.dsmb)
seqPlotPower(sepsis.dsmb2, sepsis.dsmb, reference="sepsis.dsmb")
seqPlotASN(sepsis.dsmb2, sepsis.dsmb, sepsis.fix, fixed=F)

#SPONSOR FUTILITY BOUNDARY
sepsis.futility <- update(sepsis.dsmb2, P=c(1,.9))
print(sepsis.futility)
```

```

sepsis.futility2 <- update(sepsis.dsmb2, P=c(1,.75))
sepsis.futility3 <- update(sepsis.dsmb2, P=c(1,.5))

seqPlotBoundary(sepsis.dsmb2, sepsis.futility, sepsis.futility2,
                sepsis.futility3, superpose.design=T)
seqPlotASN(sepsis.dsmb2, sepsis.futility, sepsis.futility2,
            sepsis.futility3, fixed=F)
seqPlotPower(sepsis.dsmb2, sepsis.futility, sepsis.futility2, sepsis.futility3)
seqPlotPower(sepsis.futility, sepsis.futility2, sepsis.futility3,
            sepsis.dsmb2, reference="sepsis.dsmb2")

#ADVANCED BOUNDARY SCALES
changeSeqScale(sepsis.futility2, "C")
changeSeqScale(sepsis.futility2, seqScale("C", hypTheta="estimate"))
changeSeqScale(sepsis.futility2, seqScale("H", priorTheta=0,
                                         priorVariation=Inf))
changeSeqScale(sepsis.futility2, seqScale("B", priorTheta=-.04,
                                         priorVariation=.1))

```

changeSeqScale Change Scale for Sequential Statistics

DESCRIPTION

Converts group sequential test statistics from one scale to another.

```
changeSeqScale(x, outScale, analysis.index=NULL, observed=NULL,  
               inScale = attr(x$boundary,"scale"))
```

REQUIRED ARGUMENTS

x: group sequential design object, inheriting from class "**seqDesign**".

outScale: scale on which to report converted statistics. Both **inScale** and **outScale** are "**seqScale**" objects or equivalent character strings, see **seqScale**.

OPTIONAL ARGUMENTS

analysis.index: a vector of analysis times at which observations are made.

observed: a vector of observed statistics, the same length as **analysis.index**.

inScale: scale on which the **observed** values are measured. The default value is the display scale for **x**.

VALUE

If **analysis.index** and **observed** are not supplied, a **seqBoundary** object on the desired scale is returned. Otherwise, a data frame with variables:

analysis.index: the input.

observed: the input.

outValue: the converted group sequential statistics.

In addition, the data frame has attributes

inScale: the input **inScale**

outScale: the input **outScale**

design: the input **x**.

DETAILS

The **outValue** is missing (NA) for (**analysis.time**, **observed**) pairs that are in continuation region (i.e. values that do not cause stopping).

Both **seqBoundary()** and **changeSeqScale()** can be used to convert stopping boundaries between scales. Only **changeSeqScale()** can be used to convert group sequential statistics that are not on a boundary.

For most scales, the results obtained for the two functions are identical. The exception is the error spending scale. The differences between the two functions for this scale is fairly technical. `seqBoundary()` returns the error spending function for the boundary by taking into account the continuation sets at each stopping time. `changeSeqScale()` does not consider continuation sets at the analysis time corresponding to the observed statistic being converted. The typical user will be more interested in the output of `seqBoundary()` for the error spending function, and the conversion of a statistic not on the boundary is generally not of as much interest on this scale.

As a general rule, `changeSeqScale()` has less computational overhead and is faster than `seqBoundary()`.

SEE ALSO

`seqBoundary`, `seqScale` .

EXAMPLE

```
des <- seqDesign(prob.model = "proportions",
                   sample.size = c(90, 130, 180),
                   conf.level = .95, P = 1, null.hypothesis = 0.25)
seqBoundary(des)                      # "X" scale by default
seqBoundary(des, "C")                 # boundaries on conditional futility scale
changeSeqScale (des, "C")            # boundaries on conditional futility scale
observed <- c(12/45 - 14/45, 19/65 - 25/65)
changeSeqScale (des, "Z", c(1,2), observed=observed)
```

checkSeqBoundary Check validity of a boundary object

DESCRIPTION

Checks if a group sequential boundary object has the right structure and has valid boundaries

```
checkSeqBoundary(x, theta.is.exp = F, NAOK = T)
```

REQUIRED ARGUMENTS

x: a boundary object or an object which can be coerced as a boundary object.

OPTIONAL ARGUMENTS

theta.is.exp: logical, if TRUE then all values must be positive for some scales.
x.

NAOK: if FALSE then any missing values cause the result to be FALSE.

VALUE

logical value, FALSE if problems are detected – values of out range, or missing values.

SEE ALSO

`seqBoundary, seqBoundary.object`

checkSeqScale Checks if display scale is valid

DESCRIPTION

Checks if a display scale is valid, and convert to a simplified form. This function is intended for internal use, by other SeqTrial functions.

```
checkSeqScale(scale, display = T, theta.is.exp = F,  
              invalid = NULL)
```

REQUIRED ARGUMENTS

scale: a **seqScale** object, or a character string that corresponds to one of the scales (listed below) that do not require parameters.

OPTIONAL ARGUMENTS

display: logical, if TRUE then perform checks as if the parameter will be used for displaying boundaries, if FALSE perform additional checks to ensure validity for additional uses of the scale; see **seqScale**.

theta.is.exp: logical, if TRUE then certain scale parameters must be positive (e.g. prior medians for Bayesian scales).

invalid: character vector, indicating scale types which should be considered invalid. This argument should depend on the intended use of the object.

VALUE

if scale is invalid (for the desired use) then character(0); otherwise one of the characters "S", "X", "Z", "P", "E", "B" or "St", whichever is matched by **scale**.

DETAILS

This function first checks whether **scale** matches one of the characters just listed, or if it partially matches one of "partial.sum", "sample.mean", "z.value", "p.value", "error", "bayes" or "standardized". If not, then **scale** does not represent a valid scale, and a character string of length 0 is returned.

If **scale** matches any string in **invalid**, the scale is considered invalid.

Otherwise validity checks are performed; For some of the scales, scale parameters are required; in these cases **scale** must be a **seqScale** object with the corresponding parameters in its **parameters** attribute. The parameters must satisfy certain conditions, depending on the values of **display** and **theta.is.exp**. If all checks are satisfied, one of the characters listed earlier is returned.

SEE ALSO

`seqScale`, `seqScale.object`.

EXAMPLE

```
checkSeqScale("X")

# checkSeqScale("B") # Would fail, because the
#                                Bayesian scale requires parameters
#                                that describe the prior.
```

confidenceInterval.seqDesign Confidence interval for a group sequential design

DESCRIPTION

Computes a confidence interval for a statistic based on a group sequential design.

```
confidenceInterval.seqInference(object)
confidenceInterval.seqDesign(object, ...)
confidenceInterval.seqDesign(object,
    analysis.index, observed,
    conf.level = .95, ordering = "b", labels, ...)
```

REQUIRED ARGUMENTS

object: a group sequential design object inheriting from "seqDesign", or an group sequential inference object inheriting from "seqInference".

OPTIONAL ARGUMENTS

....: additional arguments which are passed to the **seqInference** function. These include:

analysis.index: a numeric vector specifying the analysis times associated with **observed**.

observed: a numeric vector specifying the observed value of the statistic. **analysis.index** and **observed** must have the same length; if both are NULL then the borders of the design are used as observed values.

conf.level: the confidence level of the confidence intervals. Intervals produced are two-sided; to obtain one-sided intervals let **conf.level** equal 2*(desired one-sided confidence) - 1, and ignore one side of the two-sided intervals.

ordering: a character indicating how to compute the confidence intervals: "m" corresponds to sample mean ordering, "a" corresponds to analysis time ordering, and "b" to both; see **seqInference**.

labels: character vector the same length as **observed**, for printing results.

VALUE

a data frame of class c("seqConfidenceInterval", "data.frame") which has columns

analysis.index:

observed:

```
CIlo.meanOrder:  
CIhi.meanOrder:  
CIlo.timeOrder:  
CIhi.timeOrder:  
labels: and certain attributes; see seqInference.object  
for descriptions of the columns and attributes.
```

DETAILS

This function calls **seqInference** (unless **object** is already a "**seqInference**" object), then extracts selected columns from the result and assigns a new class. There is a print method for this class.

SEE ALSO

seqInference, **seqInference.object** , **seqDesign** .

EXAMPLE

```
des <- seqDesign(sample.size = c(60, 90, 120, 150), P = .5)  
confidenceInterval(des, analysis.index = 2, observed = .24)  
seqInference(des, analysis.index = 2, observed = .24)
```

confidenceInterval Confidence interval

DESCRIPTION

Compute confidence intervals

This function is an S Version 3 generic (see Methods); method functions can be written to handle specific S Version 3 classes of data. Classes which already have methods for this function include:

seqDesign seqInference

confidenceInterval(object, ...)

REQUIRED ARGUMENTS

object: an object which is dispatched to a method. There is currently no default method.

OPTIONAL ARGUMENTS methods may have additional arguments.

VALUE

an object representing confidence intervals.

SEE ALSO

confidenceInterval.seqDesign, confidenceInterval.seqInference
, seqDesign , seqInference .

confidenceLevel The confidence level for a hypothesis test

DESCRIPTION

Extract the confidence level for a hypothesis test.

This function is an S Version 3 generic (see Methods); method functions can be written to handle specific S Version 3 classes of data. Classes which already have methods for this function include:

seqHypothesis seqDesign

```
confidenceLevel(x)
```

REQUIRED ARGUMENTS

x: an object inheriting from the class "seqHypothesis" or the class "seqDesign".

VALUE

the confidence level of the test.

SEE ALSO

seqDesign, seqHypothesis.object

EXAMPLE

```
des <- seqDesign(sample.size = c(20,40,60,80))
# the default is a .975 level one-sided test
confidenceLevel(des)
```

dSeq Distribution of Test Statistic in Group Sequential Trial

DESCRIPTION

Density, cumulative probability, quantiles, and expected value for a test statistic in a group sequential trial.

```
dSeq(x, analysis.index, observed, theta,
      inScale = attr(x$boundary, "scale"))
pSeq(x, analysis.index, observed, theta,
      inScale = attr(x$boundary, "scale"), ordering="b")
qSeq(x, prob, theta, outScale="X", ordering="b")
meanSeq(x, theta)
```

REQUIRED ARGUMENTS

x: group sequential design object, inheriting from class "**seqDesign**".

analysis.index: a vector of analysis times at which observations are made.

observed: a vector of observed statistics, the same length as **analysis.index**.

prob: vector of probabilities for which quantiles are desired.

theta: scalar, or vector the same length as **analysis.index** or **prob**, or vector of arbitrary length for **meanSeq** giving values of theta to be used.

OPTIONAL ARGUMENTS

inScale: scale on which the **observed** values are measured. Both **inScale** and **outScale** are "**seqScale**" objects or equivalent character strings, see **seqScale**.

outScale: scale on which to report quantiles.

VALUE

dSeq and **pSeq** return data frames with variables:

analysis.index: the input.

observed: the input.

theta: the input, replicated if necessary to the right length.

dens: density (**dSeq** only).

cdf.meanOrder: cumulative distribution function (**pSeq** only).

cdf.timeOrder: cumulative distribution function (**pSeq** only).

In addition, both have attributes

scale: the input `inScale`
design: the input `x`. `qSeq` returns a data frame with variables:
prob: the input.
theta: the input, replicated if necessary to the right length.
qObsMean: quantiles of the sample mean (using mean ordering)
analysis.index, observed: (time,value) pairs, quantiles of the distribution, using analysis time ordering.
The data frame has attributes:
scale: the input `outScale`
design: the input `x`. `qSeq` returns a data frame with variables:
theta: the input.
expSampMean: the expected sample mean given `theta`.
and attribute
design: the input `x`.

DETAILS

The density is zero for (`analysis.time, observed`) pairs that are in continuation region (i.e. values that do not cause stopping).
The cumulative distribution (`pSeq`) and quantiles (`qSeq`) are calculated using both analysis time ordering and mean ordering; These determine how observed values at different times are ordered.

REFERENCES

See references for `seqInference`.

SEE ALSO

`seqDesign, seqInference`.

EXAMPLE

```

des <- seqDesign(prob.model = "proportions",
                  sample.size = c(90, 130, 180),
                  conf.level = .95, P = 1, null.hypothesis = 0.25)
observed <- c(12/45 - 14/45, 19/65 - 25/65)
dSeq(des, c(1,2), observed = observed, theta = 0.0)
pSeq(des, c(1,2), observed = observed, theta = 0.0)
qSeq(des, c(.025, .975), theta = 0.0)
meanSeq(des, theta = c(-0.1, 0.0, 0.1))

```

is.seqHypothesis Test if an object is a seqHypotheses object

DESCRIPTION

Test if an object is a seqHypotheses object

`is.seqHypothesis(x)`

REQUIRED ARGUMENTS

`x`: any object

VALUE

TRUE if the object inherits from class "seqHypothesis".

DETAILS

There is no `seqHypothesis` function; objects of this class are created by `seqDesign`, as the `seqHypothesis` component of a `seqDesign` object.

SEE ALSO

`seqDesign`, `seqHypothesis.object`.

EXAMPLE

```
fxd <- seqDesign(sample.size = 850, arms = 2,
                  conf.level = .99, variance = c(.25, .13))
## now extract the hypothesis
fxd$hypothesis
is.seqHypothesis(fxd$hypothesis) # Returns T (TRUE)
```

NAto0 Replace NA's with 0

DESCRIPTION

Set elements of an object which are **NA** to zero.

NAto0(x)

REQUIRED ARGUMENTS

x: a numeric valued vector, matrix, or array.

VALUE

the object, with NAs replaced by 0.

DETAILS

NAs are replaced by 0.

panel.seqBoundary.superpose Panel Function for Superposing Boundaries

DESCRIPTION

Trellis panel function for displaying the decision boundaries for one or more group sequential design on a single panel

```
panel.seqBoundary.superpose(x, y,
    subscripts, groups, vertical.lines = T,
    ...,
    jitter = <><see below>>)
```

REQUIRED ARGUMENTS

x: the sample size corresponding to the boundaries.

y: the decision boundaries.

subscripts: vector used for indexing, indicates which values of **x** and **y** should be plotted.

groups: factor or numeric vector indicating groups; one set of boundaries is graphed for each group.

OPTIONAL ARGUMENTS

vertical.lines: if TRUE, vertical lines will be drawn connecting the stopping boundaries to the edges of the plotting region.

....: other graphical parameters can be passed to **panel.seqBoundary**; see **par**. By default, the trellis parameter list **superpose.line** will be used to determine line width (**lwd**), line type (**lty**), and color (**col**). The trellis parameter list **superpose.symbol** will be used to control character size (**cex**), plotting symbol (**pch**), and font.

jitter: scalar or vector, the amount to jitter vertical lines for each different design. Jittering makes it possible to distinguish multiple designs with decision boundaries at the same sample size. The default is **0:(max.groups-1) * diff(range(x)) / 200**. If **jitter** is scalar then it is reset to **0:(max.groups-1) * jitter**. Jittering is only applied at analysis times which are shared by at least two designs.

A plot of the boundaries for one or more designs is displayed on the same panel.

DETAILS

The lines connecting the boundaries are plotted with the same line parameters as the decision boundary lines.

This function is internal to the SeqTrial module and is not intended for direct use. This function is used by **seqPlotBoundary**.

SEE ALSO

seqPlotBoundary. The following are located in S-PLUS Help: panel.superpose,
par.

panel.seqBoundary Panel Function for Decision Boundaries

DESCRIPTION

Trellis panel function for displaying the decision boundaries for a group sequential design.

```
panel.seqBoundary(x, y, vertical.lines = T,
                  observed = NULL, N = NULL,
                  ....,
                  connect.lty = <<see below>>,
                  connect.lwd = <<see below>>,
                  connect.col = <<see below>>,
                  jitter = 0)
```

REQUIRED ARGUMENTS

x: the sample size corresponding to the boundaries.

y: the decision boundaries.

OPTIONAL ARGUMENTS

vertical.lines: if TRUE, vertical lines will be drawn connecting the stopping boundaries to the edges of the plotting region.

observed: NULL, or a vector of observed statistics on the same scale as the boundaries.

N: vector the same length as **observed**; the sample size corresponding to **observed**.

....: other graphical parameters can be passed to **points** or **lines**; see **par**. By default, the trellis parameter list **plot.line** will be used to line width (**lwd**) and line type (**lty**). The trellis parameter list **superpose.symbol** will be used to control character size (**cex**), plotting symbol (**pch**), font, and color (**col**). The stopping boundaries are plotted using **cex[1]**, **pch[1]**, **font[1]**, and **col[1]**. If **observed** and **sample.size** are provided, then they will plotted using **cex[2]**, **pch[2]**, **font[2]**, and **col[2]**.

connect.lty: the trellis parameter **lty** used to plot the lines which connect the boundaries. By default, **connect.lty = trellis.par.get("dot.line")\$lty**

connect.lwd: the trellis parameter **lwd** used to plot the lines which connect the boundaries. By default, **connect.lwd = trellis.par.get("dot.line")\$lwd**

connect.col: the trellis parameter **col** used to plot the lines which connect the boundaries. By default, **connect.col = trellis.par.get("dot.line")\$col**

jitter: scalar; if vertical lines are drawn, they are shifted to the the right by this amount.

An plot of the boundaries for a single design is displayed.

DETAILS

This function is internal to the SeqTrial module and is not intended for direct use. This function is used by **seqPlotBoundary**.

SEE ALSO

seqPlotBoundary, **panel.seqBoundary.superpose** , **par** (located in S-PLUS Help).

panel.seqInference Panel Function for Inferences

DESCRIPTION

Trellis panel function for displaying the inferences for a group sequential design.

```
panel.seqInference(x, y, ...,
  whisker = diff(range(x))/100, digits = 3,
  col.text = trellis.par.get("add.text")$col,
  font.text = trellis.par.get("add.text")$font,
  col.hypothesis = trellis.par.get("reference.line")$col,
  lty.hypothesis = trellis.par.get("reference.line")$lty,
  lwd.hypothesis = trellis.par.get("reference.line")$lwd)
```

REQUIRED ARGUMENTS

x: the sample size corresponding to the inferences.

y: the inferences.

....: other graphical parameters can be passed to **segments** for the confidence intervals, or **points** for the observed data and point estimate; see **par**.

whisker: the width of the whiskers at the ends of the segments displaying the confidence intervals.

digits: number of digits to use for rounding when creating labels.

col.text: the color of the p-values to be printed as text.

font.text: the font of the p-values to be printed as text.

col.hypothesis: the color of the null and alternative hypotheses which are superposed as horizontal lines on the panel.

lty.hypothesis: the line type of the null and alternative hypotheses which are superposed as horizontal lines on the panel.

lwd.hypothesis: the line width of the null and alternative hypotheses which are superposed as horizontal lines on the panel.

A plot of the inferences for a single design is displayed.

DETAILS

This function is internal to the SeqTrial module and is not intended for direct use. This function is used by **seqPlotInference**.

SEE ALSO

seqPlotInference, **par** (located in S-PLUS Help).

panel.seqPlotPower Panel Function for Power Curves

DESCRIPTION

Trellis panel function for displaying power curves for a group sequential design.

```
panel.seqPlotPower(x, y, ..., reference = F, superpose = F)
```

REQUIRED ARGUMENTS

x: the natural parameter theta at which power is computed.

y: the power curve defined by defined by

$$\Pr(S[M] \leq a[M] \mid \theta)$$

(lower power) or by

$$\Pr(S[M] \geq d[M] \mid \theta)$$

(upper power), where M is the final stopping time and $S[M]$, $a[M]$ and $d[M]$ are the values of the statistic, lower, and upper boundaries, respectively.

OPTIONAL ARGUMENTS

....: other graphical parameters can be passed to **panel.xyplot** or **panel.superpose**.

reference: if TRUE, then a reference line will be added at zero. The trellis parameter list **reference.line** is be used to control the reference line.

superpose: if TRUE, the **panel.superpose** is called. Otherwise **panel.xyplot** is called.

A plot of the power curves is produced.

DETAILS

This function is internal to the SeqTrial module and is not intended for direct use. This function is used by **seqPlotPower**.

SEE ALSO

seqPlotPower.

panel.seqPlotStopProb Panel Function for Stopping Probabilities

DESCRIPTION

Trellis panel function for displaying the stopping probabilities for a group sequential design.

```
panel.seqPlotStopProb(x, y,
  subscripts, groups,
  ..., col.points = trellis.par.get("add.text")$col
  addPoints = T)
```

REQUIRED ARGUMENTS

x: the sample size corresponding to the stopping probabilities.

y: the cumulative stopping probabilities.

subscripts: vector used for indexing, indicates which values of **x** and **y** should be plotted.

groups: factor or numeric vector indicating groups; curve of stopping probabilities is produced for each group.

OPTIONAL ARGUMENTS

...: other graphical parameters can be passed to **polygon** or **points**; see **par**.

col.points: the color of the points used to demark the total stopping probability at each analysis time.

addPoints: logical, if TRUE then numbers indicating the analysis time are added at the cumulative values for each time.

A plot of the stopping probabilities for a single design is displayed.

DETAILS

This function is internal to the SeqTrial module and is not intended for direct use. This function is used by **seqPlotStopProb**.

SEE ALSO

seqPlotStopProb. The following are located in S-PLUS Help: polygon, points, par.

plot.seqOperatingChar Plot Operating Characteristics

DESCRIPTION

Plot operating characteristics (power, stopping probabilities, or expected sample size for a group sequential design.

```
plot.seqOperatingChar(x, ..., which = "power")
```

REQUIRED ARGUMENTS

x: a `seqDesign` object, a list of `seqDesign` objects, a `seqOperatingChar` object, or a list of `seqOperatingChar` objects.

OPTIONAL ARGUMENTS

...: additional arguments which are passed to one of `seqPlotPower`, `seqPlotASN`, `seqPlotStopProb`.

which: character; choices are "power", "ASN", and "stopping.prob". This determines which of the three functions is called. Partial matching is used, so e.g. "stop" is sufficient.

VALUE

an object of class `trellis` which is automatically plotted by `print.trellis`.

DETAILS

This function is a front end to the other three functions.

SEE ALSO

`seqPlotPower`, `seqPlotASN` , `seqPlotStopProb` .

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P = 1)
asym1 <- update(symm, P = c(.9,1))
asym2 <- update(symm, P = c(.7,1))
oc3 <- seqOperatingChar(list(symm, asym1, asym2))
plot(oc3) # power
plot(oc3, which = "ASN")
plot(oc3, which = "stop")
```

powerTest The power for a hypothesis test

DESCRIPTION

Extract the power for a hypothesis test.

This function is an S Version 3 generic (see Methods); method functions can be written to handle specific S Version 3 classes of data. Classes which already have methods for this function include:

seqHypothesis seqDesign

powerTest(x)

REQUIRED ARGUMENTS

x: an object inheriting from the class "seqHypothesis" or the class "seqDesign".

VALUE

the power of the test at the lower and upper alternatives.

SEE ALSO

seqDesign, seqHypothesis.object , seqExtract .

EXAMPLE

```
des <- seqDesign(sample.size = c(20,40,60,80))
# the upper and lower power is .975
powerTest(des)
```

sampleSize Sample Size

DESCRIPTION

Extract the sample size for a hypothesis test

This function is an S Version 3 generic (see Methods); method functions can be written to handle specific S Version 3 classes of data. Classes which already have methods for this function include:

seqDesign seqParameters

```
sampleSize(x)
```

REQUIRED ARGUMENTS

x: an object inheriting from the class "seqDesign".

VALUE

the sample size for a hypothesis test, total number of observations (for all arms or groups).

DETAILS

For hazard models, sample size refers to the total number of events to be observed in the study (not including censored observations). See **seqPHSubjects**

to estimate the number of observations necessary to obtain the specified number of events.

SEE ALSO

seqDesign, seqFixDesign , seqExtract , seqPHSubjects .

EXAMPLE

```
des <- seqDesign(sample.size = c(20,40,60,80))
sampleSize(des)
des <- seqDesign(alt.hypothesis = .1, nbr.analyses = 3)
sampleSize(des)
```

seqBoundaryLabels Label for the boundary scale

DESCRIPTION

Returns a label suitable for printing or plotting for the boundary scale.

`seqBoundaryLabels(type)`

REQUIRED ARGUMENTS

type: one of the characters "S", "X", "Z", "P", "E", "B" or "St" or a string which partially matches "partial.sum", "sample.mean", "z.value", "p.value", "error.spend", "bayes" or "standardized".

VALUE

if **type** matches, then a descriptive label is returned.

seqBoundary.object Decision Boundary Object

DESCRIPTION

Objects which inherit from **seqBoundary**.

GENERATION

Objects which inherit from "seqBoundary" are returned from the **seqBoundary** and **as.seqBoundary** functions.

METHODS

The class "seqBoundary" has **plot** and **print** methods.

OTHER FUNCTIONS

The following functions are related to

WARNING

manipulating "seqBoundary" objects:

seqBoundaryTimeValue, **seqBoundaryLabels**, **checkSeqBoundary** ,
and **checkSeqScale**.

VALUE

The "seqBoundary" object is a matrix with K rows and 4 columns, where K is the number of analysis times. The columns of the matrix correspond to the boundaries "a", "b", "c", and "d" respectively, and the rows to analysis times.

A **seqBoundary** object has attributes:

sample.size: vector of analysis times.

scale: a "seqScale" object, indicating the scale on which the boundaries are specified.

no.stopping: logical vector of length K, indicating at which analysis times the internal ("b" and "c") boundaries are inactive.

SEE ALSO

seqBoundary, **seqDesign** , **seqDesign.object** , **seqScale** , **seqScale.object**

seqBoundary Create a Boundary Object

DESCRIPTION

Create a boundary object which determines the stopping rules for a group sequential test.

```
seqBoundary(x, scale, sample.size, no.stopping, ...)
as.seqBoundary(x, ...)
is.seqBoundary(x)
```

REQUIRED ARGUMENTS

x: an object of class "**seqDesign**", an object of class "**seqBoundary**", or a matrix with K rows and 4 columns, where K is the number of analysis times and the columns correspond the boundaries "a", "b", "c", "d".

OPTIONAL ARGUMENTS

scale: an object of class "**seqScale**", or a character string which may be converted to a "**seqScale**" object.

sample.size: vector of analysis times, of length K. This is ignored if **x** is a "**seqDesign**" object.

no.stopping: a logical vector of length K, indicating which analysis times have inactive internal ("b" and "c") boundaries, or a matrix (see details below). This is ignored if **x** is a "**seqDesign**" object. By default it is determined from **x** (this is not possible for all scales).

...: additional argument which are passed to the **seqScale** function, and are used to convert **scale** to a **seqScale** object.

VALUE

a boundary object, of class "**seqBoundary**". This is a matrix with K rows and 4 columns, and certain attributes. See **seqBoundary.object** for details. Also see **seqScale.object** for details on possible scales.

DETAILS

There are two basic uses for this function, depending on **x**. First, if **x** inherits from class "**seqDesign**". If no **scale** is specified, then **x\$boundary** is returned; Otherwise the boundaries are converted to the scale specified by **scale**.

The **changeSeqScale** function can also change boundaries for a "**seqDesign**" object, is generally faster, and is usually equivalent. See **changeSeqScale** for further details.

Second, if `x` is a "seqBoundary" object or other matrix, then the object is converted to a "seqBoundary" object with the specified `scale` and `sample.size`; for a seqBoundary object these default to the old values.

The `no.stopping` argument may also be another numeric matrix (or seqBoundary object) with `K` rows and 4 columns; the necessary logical vector is then calculated according to whether the inner boundaries are active if this matrix were used as a boundary.

SEE ALSO

`changeSeqScale`, `seqBoundary.object`, `seqDesign`, `seqDesign.object`,
`seqScale`, `seqScale.object`.

EXAMPLE

```
## Custom boundaries for a design with 2 analysis times
bound <- seqBoundary(rbind(c(-1.645, 0, 0, Inf),
                           qnorm(c(.0225, .0225, .9775, .9775))),
                      scale = "Z")
des <- seqDesign(sample.size = 850, nbr.analyses = 2,
                  exact.constraint = bound,
                  test.type = "two.sided")
## extract the boundary on various scales
seqBoundary(des, scale = "S") # partial sum scale
seqBoundary(des, scale = "P") # fixed-sample P-value
```

seqBoundaryTimeValue Convert a boundary object to time-value pairs

DESCRIPTION

Converts a group sequential boundary object to time-value pairs dropping boundaries corresponding to non-decision intervals.

seqBoundaryTimeValue(bnd)

REQUIRED ARGUMENTS

bnd: an object of class "seqBoundary".

VALUE

a matrix with P rows and 2 columns where P is the number of boundaries, excluding the boundaries which are **Inf** and **NA** and excluding "**b**"-"**c**" boundaries corresponding to non-decision intervals. The first column is the analysis time and the second column is the boundary stopping value.

DETAILS

This function is used to produce a list of boundaries at which to compute inferences.

SEE ALSO

seqBoundary, **seqBoundary.object** , **seqInference** .

seqDesign.boundaryShape seqDesign Boundary Shape Details

DESCRIPTION

Details on arguments to **seqDesign** related to boundary shapes.

```
seqDesign(..., design.family = "X",
          P = 1, R = 0, A = 0, G,
          early.stopping, exact.constraint,
          minimum.constraint, maximum.constraint)
```

....: other arguments to **seqDesign**.

design.family: This may be a character string, indicating one of the special design families found in the literature; see **seqDesign.specialFamilies**. Those special families imply certain values for the shape parameters (P, R, and A).

Alternately, **design.family** may be a **seqScale** object, or a character string that corresponds to a **seqScale** object without additional parameters. This scale affects how the P, R, A, and G parameters (and possibly constraint matrices) are interpreted. Possible scales are: "X" (sample mean), "S" (partial sum), "Z" (Z value), "E" (error spending – accepts additional parameters, see **seqScale**).

P, R, A, G: vectors of length 4, giving shape parameters for the a, b, c, and d boundaries. (P, R, and A may also be length 1 or 2, see below.). The basic form of each boundary shape is

$$y = \theta + (A + t\hat{A}(-P)(1-t)\hat{R})G$$

on the scale indicated by **design.family**, where **theta** depends on the hypotheses and test type, and **t** = (sample size)/(maximum sample size). For further details see the "S+SeqTrial Technical Overview".

If **design.family** corresponds to a special family, then the shape parameters are set accordingly. Otherwise the default values correspond to an O'Brien-Fleming boundary shape: R = A = 0, and P is 1, 0, 0.5, or -3.25 for **design.family** equal to "X", "S", "Z", "E", respectively. For the error spending family ("E") these defaults provides a very close approximation to the Type I error spending function for a two-sided level .05 O'Brien - Fleming boundary shape function.

R must be nonnegative. For the error spending family, A is ignored, and an error occurs if Pgt;0 or both Plt;0 and Rgt;0 for any boundary.

Arguments P, R, and A may be vectors of length 1 or 2:

`length==1`

this value is used for all boundaries that allow early stopping (as dictated by `early.stopping`).

`length==2`

the two values of `P` (or `R` or `A`; similarly below) specify lower and upper boundary shapes when only two boundaries are used, otherwise inner and outer shapes. In particular:

For one-sided hypothesis tests, (`test.type="greater"`, `"less"`, or `"equivalence"`), `P` gives the `a` and `d` boundaries (`b` and `c` are not used).

For other tests with `early.stopping="alternative"` `P` gives the `a` and `d` boundaries, (there is no early stopping for `b` and `c`).

For other tests with `early.stopping="null"` `P` gives the `b` and `c` boundaries, (there is no early stopping for `a` and `d`).

For other tests with `early.stopping="both"`, `P[1]` gives both inner (`b` and `c`) and `P[2]` gives both outer (`a` and `d`) boundaries.

G: typically this argument is not specified, but is calculated by the function. If specified, a vector of length 4 corresponding to the `a`, `b`, `c`, and `d` boundaries, respectively, to be used as starting values in the search for a group sequential design.

`early.stopping`: a character string, one of: `"null"`, `"alternative"`, or `"both"`, indicating for which hypotheses early stopping is allowed. The default value is `"both"` for one-sided tests, and `"alternative"` otherwise. This argument is ignored if any of `P`, `R`, or `A` are specified for two boundaries for a one-sided hypothesis test or for four boundaries for a two-sided test. For equivalence tests or other shifted one-sided tests this must be set to `"both"`.

`exact.constraint`: a `seqBoundary` object or ordinary matrix with one column for each of the four boundaries and one row for each analysis time. Or, a `seqDesign` object that contains an `exact.constraint`. This specifies boundaries which must be matched exactly; a missing value (`NA`) in any position in the matrix indicates there is no exact constraint for that boundary at that time. If this is an ordinary matrix then the constraints are interpreted according to the `design.family` scale; otherwise the scale contained in the `seqBoundary` object is used. Possible scales include the same choices `"X"`, `"S"`, `"Z"`, `"E"` as for `design.family`, and additional choices:

`"P"` (fixed sample `P` value), and

`"C"` (conditional futility).

`"St"` (standardized cumulative sum),

`"B"` (Bayesian) and

`"H"` (predictive futility).

The latter two choices must be passed in the form of a `seqScale` object, to pass required parameters (e.g. parameters for the Bayesian

prior distribution). However, if `design.family == "X"` then the constraint scale may not be "E" or "St", and if `design.family == "E"` then the constraint scale must be "E".

`minimum.constraint`, `maximum.constraint`: a `seqBoundary` object or ordinary matrix like the `exact.constraint` argument, or `seqDesign` object containing a `minimum.constraint` or `maximum.constraint`, which determines minimum or maximum constraints for the boundaries, respectively.

DETAILS

The S+SeqTrial User's Manual describes boundary shape parameters in more detail, including reasonable ranges for the `A` parameter.

If `exact.constraint` is completely specified (no missing values), then other arguments to `seqDesign` may be partially or completely ignored, and results may not be self-consistent. For example, `power` may be ignored (because power is determined by the sample size and the boundaries specified by `exact.constraint`), and the actual power may differ from the nominal power found in `result$hypothesis$power`.

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials, *Biometrics*, Vol 35, 549-556.

Kittelson, J. M. and Emerson, S. (1999). A Unifying Family of Group Sequential Test Designs, *Biometrics*, Vol. 55, 874-882.

SEE ALSO

`seqDesign`, `seqDesign.specialFamilies`, `seqScale`.

seqDesignCtoS Wrapper for Interface to C Code

DESCRIPTION

Converts a group sequential design information returned from a C function into a **seqDesign** object.

```
seqDesignCtoS(clist, specification, call)
```

REQUIRED ARGUMENTS

clist: a list returned from a C function in the SeqTrial C library.

specification: list containing values of the arguments specified in the call.

call: the call which created the object.

VALUE

an object which inherits from class "**seqDesign**".

DETAILS

This function is internal to the SeqTrial module and is not intended for general usage.

SEE ALSO

seqDesign, **seqDesign.object** .

seqDesign.multiple.arms seqDesign multiple arms trials (arms > 3)

DESCRIPTION

Details on arguments to **seqDesign** related to probability models for more than two arms.

```
seqDesign(prob.model = "normal", arms, log.transform = F,  
         null.hypothesis, alt.hypothesis, variance, ratio,  
         nbr.analyses = 1, ...)
```

MULTIPLE-ARM TRIALS

This help file describes arguments with particular meanings for trials with multiple (three or more) arms. Only fixed sample (**nbr.analyses** = 1) study designs with **prob.model** = "normal" or **prob.model** = "lognormal" are currently supported for multiple arm trials.

OPTIONAL ARGUMENTS

prob.model: a character string specifying the probability model used for the primary measure of treatment outcome. For trials with more than two arms, the only current choices are "normal" and "lognormal".

prob.model="normal"

(the default) Analyses are based on the usual normal distribution based theory for a continuously distributed measure of response. Outcome on each treatment arm is the mean response, and the overall summary is the between-group variance of the means for the arms. Variances of the underlying distributions must be equal in each arm.

prob.model="lognormal"

Analyses are based on the lognormal distribution for a positively skewed, continuously distributed positive measure of response. Outcome on each treatment arm is the median (geometric mean) response, and the overall summary is the between-group variance of the logs of the medians for the arms. Variances of the underlying distributions must be equal in each arm after a log transformation.

arms: a nonnegative integer value specifying the number of treatment arms for the trial. All text in this help file is for the case that **arms** > 3.

log.transform: logical value. if **prob.model** == "lognormal" this affects argument **alt.hypothesis**, see below.

The meaning of this argument is different for multiple-arm trials than in the rest of SeqTrial (see **seqDesign.probabilityModel**).

null.hypothesis: scalar, this value must be 0 indicating zero variance between arms of sample means or log-sample medians.

alt.hypothesis : scalar, or vector of length equal to the value of **arms**. This argument is used either to specify the alternative hypothesis to use when calculating sample size or power.

The model is a fixed-effects model, with between-arms variance $V = \text{var}(M, \text{unbiased}=F)$, where M is the vector of population means or log-sample medians. Note that the variance is defined with a divisor of (arms) rather than $(\text{arms}-1)$.

A scalar value is interpreted as V . A vector value interpreted as M when **prob.model**="normal", from which V is calculated.

In the vector case with **prob.model**="lognormal", if **log.transform** = FALSE (the default), then **alt.hypothesis** should contain the population medians for each arm, and a log transform is done prior to calculating variance, $M \leftarrow \log(\text{alt.hypothesis})$. **log.transform** = TRUE indicates that **alt.hypothesis** contains the log-medians.

In the case of unequal sample sizes on the arms, the variance is calculated as the weighted variance $\sum(w * (M - \sum(w*M))^2)$ where $w = \text{ratio}/\sum(\text{ratio})$.

alt.hypothesis can also be unspecified, or the character string "calculate", signifying that **seqDesign()** should calculate the alternative that a study with sample size specified by **sample.size** can detect with power specified by **power**.

variance : scalar, the variance of an individual observation (after a log transformation in the "lognormal" case). The same variance is used for all arms.

ratio : vector of positive values, the ratio of relative sample sizes to be accrued to the trial arms. This is replicate to length **arms** if necessary. The default value is **ratio** = **rep(1,arms)**.

... : additional arguments to **seqDesign**.

VALUE

a **seqDesign** object. The **std.bounds** component of the result contains chi-square critical values, and the **boundary** component of the result contains critical values on the weighted variance scale, i.e. rejection occurs if **var(Means, unbiased=F) > result\$bounds[1]** in the case of equal allocations between arms.

DETAILS

Size and power are based on chi-square and noncentral chi-square distributions, respectively, using a fixed-effects model. Further details on the model are available in initial comments inside **seqDesignKArms**.

SEE ALSO

`seqDesign`, `seqDesign.object` , `seqDesign.probabilityModel` .

EXAMPLE

```
seqDesign(arms = 3)
seqDesign(arms=3, alt.hypothesis = 1:3) # vector
seqDesign(arms=3, alt.hypothesis = 2/3) # equivalent scalar
seqDesign(prob.model = "lognormal", arms = 3,
          alt.hypothesis = exp(1:3))    # equivalent
seqDesign(prob.model = "lognormal", arms = 3,
          log.transform = T, alt.hypothesis = 1:3) # equivalent
seqDesign(arms=3, alt.hypothesis = 2, variance = 3)
seqDesign(arms=3, alt.hypothesis = 1:3, ratio = c(1,2,1))
```

seqDesign.object Group Sequential Design Object

DESCRIPTION

Objects which inherit from **seqDesign**.

GENERATION

Objects which inherit from "seqDesign" are returned from the **seqDesign**, and **update.seqDesign** functions.

METHODS

The class "seqDesign" has methods for the following generic functions:

print, **plot**,
** Inference and Operating Characteristics:
summary, **seqInference**, **confidenceInterval**, **seqOperatingChar**,
,
** Extract Parts of the Object:
seqBoundary, **seqExtract**, **confidenceLevel**, **powerTest**, **sampleSize**

Other non-generic functions operate on the object, including **seqPlotASN**, **seqPlotInference**, **seqPlotPower**, **seqPlotStopProb**, **seqFixDesign**

VALUE

The "seqDesign" object is a list with components:

model: an object of class "seqModel", describing the probability model and hypotheses used in the design. See **seqModel.object** for details.

hypothesis: an object of class "seqHypothesis" which specifies the null and alternative hypotheses, confidence level, and power for the test. See **seqHypothesis.object** for details.

parameters: an object of class "seqParameters" containing the parameters used in creating the design, including number and times of analyses, design family (the "P", "A", "R", and "G" parameters, which determine the shapes of the boundary curves as a function of sample size), and constraints. See **seqParameters.object** for details.

boundary: a matrix with K rows and 4 columns, where K is the number of analysis times in this design. The columns of the matrix correspond to the

boundaries "a", "b", "c", and "d" respectively. This is an object of class "**seqBoundary**", see **seqBoundary.object**

for details on this and subsequent "**seqBoundary**" objects.

error.spend: an object of class "**seqBoundary**"; the columns of this matrix correspond to the error spending functions "a", "b", "c", and "d" respectively.

std.bounds: an object of class "**seqBoundary**", describing the standardized boundaries of the design.

call: the call which created the object.

date: date this design was created

specification: list containing values of the arguments specified in the call.

SEE ALSO

seqDesign, **seqBoundary.object** , **seqHypothesis.object** , **seqModel.object** , **seqParameters.object** . **summary.seqDesign** .

seqDesign.probabilityModel seqDesign Probability Model Details

DESCRIPTION

Details on arguments to **seqDesign** related to probability models.

```
seqDesign(prob.model = "normal", arms = 2, log.transform = F,  
         null.hypothesis, alt.hypothesis, variance, ratio,  
         ...)
```

OPTIONAL ARGUMENTS

prob.model: a character string (may be abbreviated) specifying the probability model used for the primary measure of treatment outcome. The "normal" and "proportions" models are additive, while the "lognormal", "odds", "poisson", and "hazard" models are multiplicative; this affects some arguments below. In this help file, "summary parameter" refers to a parameter of a single distribution, such as the mean of a normal distribution for a single arm or given a particular value of a covariate, and "treatment parameter" is the parameter that hypothesis tests are based on, e.g. a difference in two normal means in a two-arm trial.

prob.model="normal"

(the default) Analyses are based on the usual normal distribution based theory for a continuously distributed response variable. Summary parameter is the mean. By the central limit theorem, this model is appropriate for a variety of probability distributions in large samples.

prob.model="lognormal"

Analyses are based on the lognormal distribution for a positively skewed, continuously distributed positive response variable. Analyses use the usual normal distribution based theory after logarithmically transforming the data. Summary parameter is the median (geometric mean). This model is often used when the standard deviation of the response variable is proportional to the mean.

prob.model="proportions"

Analyses are based on the binomial distribution for a binary response. Statistical inference uses the normal approximation to the binomial. Summary parameter is the proportion. This model is only used in one and two armed trials.

prob.model="odds"

Analyses are based on the binomial distribution for a binary response. Statistical inference uses asymptotic normal theory.

Summary parameter is the odds of response, $p/(1-p)$ where p is the probability of success. This probability model is appropriate when the data are to be analyzed using logistic regression.

prob.model="poisson"

Analyses are based on the Poisson distribution for a response counting the number of events occurring in a defined time or region. Statistical inference uses asymptotic normal theory. Summary parameter is the mean. This probability model is appropriate when the data are to be analyzed using Poisson regression.

prob.model="hazard"

Analyses are based on the semiparametric proportional hazard model for a continuously measured time to event subject to right censoring. Inference is based on asymptotic normal theory. There is no summary parameter for each arm; arms are compared using the hazard ratio. This probability model is appropriate when the data are to be analyzed using the logrank test or Cox regression. Only two arm trials are implemented for this model.

For hazard models, sample size refers to the total number of events to be observed in the study (not including censored observations). See **seqPHSubjects**

to estimate the number of observations necessary to obtain the specified number of events.

arms: a nonnegative integer value specifying the number of treatment arms for the trial.

arms=1

a one arm trial. The treatment parameter is the summary parameter (e.g. normal mean) as specified by **prob.model**.

arms=2

(the default) a study comparing two treatments. The first arm is referred to as the treatment arm, and the second as the comparison arm. The treatment parameter is a difference (for additive models) or ratio (for multiplicative models), of the summary parameter for the treatment and control arms. E.g. for the "normal" model the treatment parameter is (mean for treatment arm) - (mean for control arm), while for the odds model it is (odds for treatment arm) / (odds for control arm).

arms>2

a multiple arm trial. This is currently implemented only for the "normal" and "lognormal" models, and some arguments (**log.transform**, **null.hypothesis**, and **alt.hypothesis**) have different meanings than described here; see **seqDesign.multiple.arms**

arms=0

a (dose-response) regression model is used, in which the response depends continuously on the value of a covariate `x` (whose distribution is specified by argument `ratio`, below). Regression models are supported for the following study designs:

`prob.model = "normal"`: usual linear regression, the treatment parameter is the slope of a linear regression of the response variable against the covariate.

`prob.model = "lognormal"`: regression, using the model $\log(\text{median}) = \beta_0 + \beta_1 x$. This corresponds to linear regression after taking a log transformation of the response. The treatment parameter is `exp(beta1)`, the ratio of medians of the lognormal response associated with changing the covariate by one unit.

`prob.model = "odds"`: Logistic regression, using the model $\log(p/(1-p)) = \beta_0 + \beta_1 x$, where `p` is the binomial proportion. The treatment parameter is `exp(beta1)`, the odds ratio associated with changing the covariate by one unit.

`prob.model = "poisson"`: Poisson regression, using the model $\log(\lambda) = \beta_0 + \beta_1 x$, where `lambda` is the Poisson mean. The treatment parameter is `exp(beta1)`, the ratio of means of the response Poisson distribution associated with changing the covariate by one unit.

`log.transform`: logical value specifying whether the summary parameter(s) and treatment parameter should be specified (see arguments `null.hypothesis` and `alt.hypothesis`) and reported on the logarithmic scale. This applies only to multiplicative models. The default is `FALSE`.

If `TRUE`, then you should specify hypotheses on a log scale, and results will be reported on the same scale. For instance, in a two arm study, the treatment parameter would be reported as the log median ratio, log odds ratio, log rate ratio, or log hazard ratio for the lognormal, binomial odds, Poisson, or proportional hazards probability models, respectively. The non-logarithmic scale may easier for nonstatisticians to understand, but statistical modeling is more natural on the log scale; e.g. in regression models (`arms=0`) the treatment parameter would be reported as `beta1` rather than `exp(beta1)` for the lognormal, logistic, and Poisson regression models.

`null.hypothesis`: numeric value (scalar or vector of length 2) specifying the null hypothesis. The valid values for this argument depends on the value of `arms` as follows:

arms=0

(a dose-response regression model) `null.hypothesis` is vector of length 1 or 2 giving regression parameters. If length 1 this gives `beta1`, the slope of the regression, as long as the model

is additive or `log.transform=T`; for multiplicative models with `log.transform=F` this gives `exp(beta1)`, the ratio between summary parameters when the covariate is changed by one unit. The default is 0.0 in the slope case, and 1.0 in the ratio case.

If length 2, `null.hypothesis[2]` is the regression slope (or ratio), while `null.hypothesis[1]` is the summary parameter (e.g. "normal" mean) for individuals with covariate equal to the average.

arms=1

`null.hypothesis` is scalar, giving the assumed summary parameter for the single arm.

The default value is `null.hypothesis = 0` for `prob.model="normal"` or multiplicative probability models when `log.transform=T`, `null.hypothesis = 0.5` when `prob.model="proportions"`, and `null.hypothesis = 1` for multiplicative probability models when `log.transform=F`. `null.hypothesis` must be positive for multiplicative probability models when `log.transform=F`, and must be between 0 and 1 when `prob.model="proportions"`.

arms=2

if `null.hypothesis` is scalar, it gives the assumed summary parameter for both arms under the null hypothesis. If a vector of length two, it gives the assumed summary parameters for treatment and comparison arms, in that order. The same defaults and constraints hold as when `arms=1`.

The exception is `prob.model = "hazard"`, where `null.hypothesis` must be a scalar value representing the treatment parameter (hazard ratio or log-hazard ratio) under the null hypothesis.

Note that except for the hazard model, you must supply the summary parameter (e.g. binomial proportion to be used for both arms, or vector of two proportions), not the treatment parameter (e.g. difference in proportions across arms); `seqDesign` needs to know the summary parameter for each arm in order to calculate variances for some models, so giving a difference or ratio of parameters is insufficient.

arms<2

See `seqDesign.multiple.arms`.

`alt.hypothesis` : this argument is used either to specify the alternative hypothesis on which the sample size and power calculations should be based, or to specify that the function should return the alternative hypothesis that the study can detect with the desired power. When a numeric value (scalar or vector of length 2) is specified, the interpretation of this argument depends on the value of `arms` as follows:

arms=0

(a dose-response regression model), a scalar value for `alt.hypothesis` is used to specify the treatment parameter (regression slope or ratio, see above) under the alternative hypothesis.

arms=1

a scalar value for `alt.hypothesis` gives the summary parameter under the alternative hypothesis.

The same constraints hold as for the null hypothesis, e.g. that proportions must be between 0 and 1.

arms=2

if `alt.hypothesis` is scalar it gives the assumed summary parameter for the treatment arm under the alternative hypothesis; the control arm is assumed to have the same value as under the null hypothesis. If `alt.hypothesis` is a vector of length two, it gives the summary parameter for the treatment and comparison arms, in that order, under the alternative hypothesis.

The exception is `prob.model = "hazard"`, where `alt.hypothesis` must be a scalar value representing the treatment parameter (hazard ratio or log-hazard ratio) under the alternative hypothesis.

Again, except for the hazard model you must supply the summary parameters, not the treatment parameter.

arms;2

See `seqDesign.multiple.arms`.

`alt.hypothesis` can also take on the following values, which indicate that a minimum detectable difference should be calculated, given specified values of `sample.size` and `power`:

alt.hypothesis="upper"

signifies that `seqDesign()` should calculate the alternative that the upper hypothesis test of a study with sample size specified by `sample.size` can detect with power specified by `power`.

alt.hypothesis="lower"

signifies that `seqDesign()` should calculate the alternative that the lower hypothesis test of a study with sample size specified by `sample.size` can detect with power specified by `power`.

alt.hypothesis="calculate"

(the default value) is equivalent to "lower" if the study design corresponds to a one-sided test of a lesser alternative, otherwise it is equivalent to "upper".

variance : This argument specifies the variability to be used in power and sample size calculations. A scalar value gives the within group variance common to all treatment levels. If `arms==2`, a vector of length 2 may

be used to specify the within group variance on the treatment and comparison arms, in that order.

prob.model="normal"

the variance of a single observation. The default value is **variance=1**.

prob.model="lognormal"

the variance of the logarithmic transformation of a single observation. The default value is **variance=1**.

prob.model="proportions"

,

prob.model="odds"

,

prob.model="poisson"

For these three models, the variability of observations under these three probability models is governed by a mean-variance relationship. However, designs are created using an approximation using a single variance, one of:

variance="null": signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under the null hypothesis.

variance="alternative": (the default) signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under the alternative hypothesis.

variance="intermediate": signifying that power or sample size calculations should be performed assuming the variability specified by the mean-variance relationship under a hypothesis intermediate to the null and alternative hypotheses.

variance= (numeric, see below)

In general, the choice **variance="null"** will give designs with the most accurate size, and **variance="alternative"** the most accurate power. **variance="intermediate"** represents a compromise, and usually gives the most accurate sample size calculations. The asymptotic approximations used in power calculations tend to be more accurate under the binomial proportions probability model than the binomial odds model.

You may specify a numeric variance for one of these three models (scalar, or optionally of length 2 if **arms==2**). This option is only recommended for experienced users. The variances should correspond to $p(1-p)$ for "proportions", $1/(p(1-p))$ for "odds", and $1/\lambda$ for "poisson", where p is a binomial proportion and λ the mean of a Poisson distribution.

prob.model="hazard"

This argument is ignored when `prob.model="hazard"` , as the variability of the test statistic is approximately proportional to one-fourth the number of events.

ratio : This numeric argument specifies the ratio of sample sizes to be accrued to the trial arms. Missing values are not allowed. The valid values for this argument depends upon the value of `arms` as follows:

arms=0

(a dose-response regression model) `ratio` indicates the distribution of levels of a covariate (a treatment variable). It is assumed that the analysis will use the regression model typically used with the specified `prob.model`, and that the predictors (which might be transformations of the treatment levels) will be modeled linearly in that model. The interpretation of `ratio` for this dose-response analysis setting will depend upon its length:

If `length(ratio)==1`, the modeled predictors are assumed to come from a normal distribution having mean 0 and variance `ratio`, default is `ratio = 1`. The variance must be positive.

If `length(ratio)==2`, the modeled predictors are assumed to come from a normal distribution having mean `ratio[2]` and variance `ratio[1]`.

If `length(ratio) > 2`, `ratio` is assumed to contain a sample from the distribution of predictors that will be observed during the trial.

arms=1

(a single arm study) a positive scalar valued `ratio` is ignored. Anything else is an error.

arms=2

a scalar value gives the ratio of subjects in the treatment arm to subjects in the control arm. A vector of length two gives relative numbers of subjects in the two arms; this is equivalent to giving the scalar value `ratio[1]/ratio[2]`.

arms_i=2

See `seqDesign.multiple.arms` .

`... : additional arguments to seqDesign.`

SEE ALSO

`seqDesign`, `seqDesign.multiple.arms` , `seqPHSubjects` .

EXAMPLE

```
seqDesign("normal", arms = 2,
          null.hypothesis = 2.5, alt.hypothesis = 2.6,
          variance = 15, ratio = 1.3)
# Equivalent; treatment group is first in various arguments
```

```

seqDesign("normal", arms = 2,
          null.hypothesis = c(2.5, 2.5), alt.hypothesis = c(2.6, 2.5),
          variance = c(15, 15), ratio = c(1.3, 1))

# Example of log.transform
seqDesign(prob.model = "hazard", alt.hypothesis = 1.1)
seqDesign(prob.model = "hazard", log.transform=T, alt.hypothesis = log(1.1))

# Specify distribution of predictors in regression case (arms=0)
#   Parameters: e.g. variance 2, mean 0:
seqDesign(prob.model = "normal", arms = 0, ratio = 2)
seqDesign(prob.model = "normal", arms = 0, ratio = c(2,0))
#   Or, give a sample of values (with distribution like you expect to the
#   real data to have (e.g. uniform between 1 and 60):
seqDesign(prob.model = "normal", arms = 0, ratio = 1:60)

```

seqDesign.specialFamilies seqDesign Special Families

DESCRIPTION

Details on special families (argument **design.family** to **seqDesign**).

seqDesign(..., design.family)

...: other arguments to **seqDesign**.

design.family: This may be a character string, one of "OBF", "Pocock", "Wang", "Symmetric", "Triangular", to indicate one of the special design families described below. Alternately, it may be a "seqScale" object, in which case other arguments to **seqDesign** determine the boundary shape; see **seqDesign** and **seqDesign.boundaryShape**.

design.family="OBF"

an O'Brien and Fleming (1979) design for a two-sided hypothesis test. The specification of this value is equivalent to making specifications as follows: **design.family=seqScale("X")**, **test.type="two.sided"**, **P=1**, **A=0**, **R=0**, **epsilon=c(1,1)**, **alpha=rep(size/2,2)**, **beta=rep(power,2)**, **early.stopping="alternative"**.

design.family="Pocock"

a Pocock (1977) design for a two-sided hypothesis test. The specification of this value is equivalent to making specifications as follows: **design.family=seqScale("X")**, **test.type="two.sided"**, **P=0.5**, **A=0**, **R=0**, **epsilon=c(1,1)**, **alpha=rep(size/2,2)**, **beta=rep(power,2)**, **early.stopping="alternative"**.

design.family="Wang"

a Wang and Tsiatis (1987) design for a two-sided hypothesis test. The specification of this value is equivalent to making specifications as follows: **design.family=seqScale("X")**, **test.type="two.sided"**, **A=0**, **R=0**, **epsilon=c(1,1)**, **alpha=rep(size/2,2)**, **beta=rep(power,2)**, **early.stopping="alternative"**. In addition, only the first value specified for P is used: **P=P[1]** .

design.family="Symmetric"

an Emerson and Fleming (1989) design for a one-sided hypothesis test or a Pampallona and Tsiatis (1994) design for a two-sided hypothesis test with equal type I and type II errors. An error results if the specification of **test.type** and **epsilon** does not correspond to either a one-sided or a two-sided design. The specification of this value is equivalent to making specifications as follows: **design.family=seqScale("X")**, **A=0**, **R=0**, **alpha=rep(size,2)** for a one-sided test or **alpha=rep(size/2,2)** otherwise, **power=1-alpha[2]**, **beta=1-alpha**, **early.stopping="both"**. In addition, only the first value specified for P is used: **P=P[1]** .

`design.family="Triangular"`

a Whitehead and Stratton (1983) design for one-sided (triangular) or two-sided (double-triangular) hypothesis test. An error results if the specification of `test.type` and `epsilon` does not correspond to either a one-sided or a two-sided design. The specification of this value is equivalent to making specifications as follows: `design.family=seqScale("X")`, `P=1` , `A=1`, `R=0` , `alpha=rep(size,2)` for a one-sided test or `alpha=rep(size/2,2)` otherwise, `beta=rep(power,2)` , `early.stopping="both"`.

REFERENCES

- Emerson, S. and Fleming, T. R. (1989). Symmetric group sequential test designs *Biometrics*, Vol 45, 905-923.
- O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials, *Biometrics*, Vol 35, 549-556.
- Pampallona, S. A. and Tsiatis, A. A. (1994). Group sequential designs for one-sided and two-sided hypothesis testing with provision for early stopping in favor of the null hypothesis. *Journal of Statistical Planning and Inference*, Vol 40.
- Pocock, S. J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika*, Vol 64, 191-199.
- Wang, S. K. and Tsiatis, A. A. (1987) Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics*, Vol 43, 193-199.
- Whitehead, J. and Stratton, I. (1983). Group Sequential Clinical Trials with Triangular Continuation Regions. *Biometrics* Vol 39, 227-236.

SEE ALSO

`seqDesign`, `seqDesign.boundaryShape` , `seqScale` .

seqDesign.testSpecifications seqDesign Test Specification Details

DESCRIPTION

Details on arguments to **seqDesign** related to specifying test size and power.

```
seqDesign(...,  
         test.type, size, conf.level = 0.95,  
         power, alpha, beta, epsilon)
```

....: other arguments to **seqDesign**.

test.type: Specification of the classical type of alternative hypothesis to be tested. This argument is ignored if the more advanced argument **epsilon** is specified. Valid values include:

test.type="greater"

(the default) a one-sided hypothesis test against a greater alternative. (This corresponds to **epsilon=c(0,1)**.)

test.type="less"

a one-sided hypothesis test against a lesser alternative. (This corresponds to **epsilon=c(1,0)**.)

test.type="two.sided"

a two-sided hypothesis test. (This corresponds to **epsilon=c(1,1)**.)

test.type="equivalence"

a one-sided hypothesis test shifted to be symmetric about 0, such as might be used in equivalence trials. (This corresponds to **epsilon=c(0.5,0.5)**.)

test.type="advanced"

a study design in which the superposition of two one-sided tests is to be specified by **epsilon**.

size: This argument is a scalar numeric value between 0 and 1 which specifies the level of significance for the hypothesis test (type I error). The default value is **size = (1 - conf.level)/2** for one-sided hypothesis tests, and the default is **size = (1 - conf.level)** otherwise.

conf.level: A numeric scalar between 0 and 1 which specifies the confidence level to be used in statistical inference. It allows easy specification of symmetric tests, i.e., tests in which the type I and type II statistical errors are equal for each of the upper and lower hypothesis tests. It leads to hypothesis testing that is equivalent to inference based on two-sided confidence intervals computed at a level equal to **conf.level**. The default value is 0.95, in which case the size of the test will be .05

for a two-sided test and .025 for a one-sided test, and the power will be .975.

power: This argument is used either to specify the desired power of the study to detect a particular alternative, or to specify that the function should return the power of the study to detect that alternative.

power=(numeric)

If **power** is a numeric scalar between 0 and 1, it specifies the desired power of the study to be used when calculating sample size or the alternative hypothesis (minimum detectable difference). If **alt.hypothesis** **lt**; **null.hypothesis** or if **alt.hypothesis="lower"**, then **power** indicates the desired power of the lower hypothesis test, otherwise the power of the upper hypothesis test.

power="calculate"

power should be calculated, based on the specified numerical values for **sample.size** and **alt.hypothesis**. The upper (lower) hypothesis test is used if the treatment effect specified by **alt.hypothesis** is greater (less) than the effect under **null.hypothesis**.

The power for the "upper" hypothesis test is the probability of falling above the upper (**d**) boundary. It does not include the probability of falling below the lower (**a**) boundary in a two-sided test (this "other-tail" probability is usually practically zero). Similarly, the power for the "lower" hypothesis test is the probability of falling below the lower (**a**) boundary.

The default value for **power** is **power=beta[1]** for a one-sided test of a lesser alternative hypothesis or if **alt.hypothesis="lower"**, and the default is **power=beta[2]** otherwise. This is equivalent to **power=1-size** for one-sided hypothesis tests and **power=1-size/2** otherwise if neither **alpha** nor **beta** is specified.

alpha: This argument allows more advanced specification of the size parameters for the group sequential design family. A vector of length two represents the level of significance for the lower hypothesis test and the level of significance for the upper hypothesis test. A scalar value represents a value to be used for the level of significance of both the lower and upper hypothesis tests. The default value is **size** for one-sided hypothesis tests and **size/2** otherwise.

If both **alpha** and **size** are specified, then **size** is ignored.

beta: This argument allows more advanced specification of the upper and lower beta parameters for the group sequential design family. A vector of length two represents the value of beta for the lower hypothesis test and the value of beta for the upper hypothesis test. A scalar value represents a value to be used for the value of beta of both the lower and upper hypothesis tests. The default value is **1-alpha**.

If both **beta** and **power** are specified, then **beta** is partially ignored; it does not affect the power at the design alternative. It may affect the shape of boundaries, depending on other parameters, and affects the **beta** and **theta.beta** components of the **hypothesis** component of the group sequential design object created by **seqDesign**. If **power** is set to a low value like 0.8, changing **beta** to the same value will often result in inferior boundaries.

epsilon: This argument allows more advanced specification of the shift of the lower and upper hypotheses. A vector of length two represents the shift of the lower hypothesis test and the shift of the upper hypothesis test. The default value is **epsilon=c(1,1)** if **test.type="two.sided"**, **epsilon=c(0,1)** if **test.type="greater"**, **epsilon=c(1,0)** if **test.type="less"**, and **epsilon=c(0.5,0.5)** if **test.type="equivalence"**.

DETAILS

Unlike fixed sample testing, it is possible to specify a family of designs in group sequential testing which span a continuum between one-sided and two-sided, using the parameter **epsilon**. One-sided tests correspond to **epsilon = c(0,1)** ("greater") and **epsilon = c(1,0)** ("less") and a two-sided test corresponds to **epsilon = c(1,1)**. An "equivalence" test is given by **epsilon = c(.5,.5)**. By adjusting **epsilon**, it is possible to obtain other intermediate designs. Moving **epsilon[1]** (or **epsilon[2]**) from 1 to 0 corresponds to deciding that it is unimportant to distinguish from the lower (upper) alternative and null hypothesis. For more details, refer to Emerson (2000).

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

SEE ALSO

seqDesign.

seqDesign Group Sequential Design

DESCRIPTION

Create a group sequential design for performing a hypothesis test.

```
seqDesign(prob.model = "normal", arms = 2, log.transform = F,
          null.hypothesis, alt.hypothesis, variance, ratio,
          nbr.analyses = 1, sample.size,
          test.type, size, conf.level = 0.95, power,
          alpha, beta, epsilon,
          design.family = "X", early.stopping,
          P = 1, R = 0, A = 0, G,
          exact.constraint, minimum.constraint, maximum.constraint,
          display.scale = "X")
update(seqDesign object, <>new arguments>>)
```

REQUIRED ARGUMENTS At least two of `power`, `alt.hypothesis`, and `sample.size` are needed. The three basic uses of this function are:

VALUE

calculate sample size: If `alt.hypothesis` is specified, and `power` is either numeric or unspecified (a default is used), then the sample size necessary to compute the power is computed. If `sample.size` is also specified, it is used only to specify the relative timing of analyses.

calculate power: If `power` is "calculate", with numerical values specified for `sample.size` and `alt.hypothesis`, then power is calculated for that alternative hypothesis and set of analysis times.

minimum detectable difference: If a numeric sample size is provided, and `power` is either numeric or unspecified (a default is used), then the alternative hypothesis which can be detected with that power and sample size is calculated.

OPTIONAL ARGUMENTS PROBABILITY MODEL ARGUMENTS ***

See `seqDesign.probabilityModel`

or `seqDesign.multiple.arms`

for details on arguments related to the underlying probability model:

`prob.model`, `arms`, `log.transform`, `null.hypothesis`, `alt.hypothesis`,
`variance`, and `ratio`.

`prob.model`: character string, one of "normal", "lognormal", "proportions",
"odds", "poisson", and "hazard".

arms: integer, number of treatment arms for the trial, usually 1 or 2. 0 indicates that treatment levels vary continuously, as in a dose-response regression model.

log.transform: logical, if TRUE then the summary measure of outcome is reported on a logarithmic scale, e.g. log odds ratio, log hazard ratio.

null.hypothesis: when **arms**=1 or **arms**=2, scalar value giving the null hypothesis value of the parameter for this probability model; for **arms**=2 may also be a vector of length 2, with control arm second. See **seqDesign.probabilityModel** for details on these cases and for **arms**=0, or see **seqDesign.multiple.arms** for **armsgt**;2.

alt.hypothesis: when **arms**=1 or **arms**=2, numeric vector of length **arms** giving the parameter under the alternative hypothesis to be used for computing sample size or power (for two-sided tests only one side is used for these computations). If **arms**=2 may be scalar; then the control value is the same as under the null hypothesis. Or, may be one of "upper", "lower", or "calculate", indicating that the minimum detectable difference should be calculated (for given **sample.size** and **power**). "calculate" is equivalent to "lower" if **test.type**="less", otherwise "upper". See **seqDesign.probabilityModel** for details on these cases and for **arms**=0, or see **seqDesign.multiple.arms** for **armsgt**;2.

variance: variance for a single observation for "normal" models; otherwise see **seqDesign.probabilityModel**.

ratio: When **armsgt**;1, relative sample sizes per arm; this is extended to length **arms** by appending 1s. When **arms**=0, this is used to specify the distribution of the predictor variable in the regression model; see **seqDesign.probabilityModel** for details.

nbr.analyses: integer, number of analysis times (interim and final). This is ignored if **sample.size** has length greater than 1.
 If **sample.size** is not a vector, then it will be assumed that the analyses are to be evenly spaced in terms of sample size. The default value (when **sample.size** is either not specified or has length 1) is 1.

sample.size: an integer or vector. If integer, gives the total number of events to be observed in the study (all treatment arms); then if **nbr.analyses** > 1, there will be that many analyses, evenly spaced in terms of sample size.
 If a vector, and one of **power** and **alt.hypothesis** are "calculate", then this gives the actual sample sizes at each interim and final analysis.
 Otherwise this gives the relative timings of the analyses – the actual sample sizes at each analysis will differ from this value by a common factor, chosen to satisfy the specified **power** at the specified **alt.hypothesis**.

*** POWER AND SIZE ARGUMENTS ***

See `seqDesign.testSpecifications`

for details about arguments related to specifying the power and size:
`test.type`, `size`, `conf.level`, `power`, `alpha`, `beta`, and `epsilon`

`test.type`: character string, one of "greater", "less", "two.sided", "equivalence", and "advanced". If "advanced" then "epsilon" is required; this is ignored if `epsilon` is specified.

`size`: scalar between 0 and 1, level of significance (Type I error). Default is $(1 - \text{conf.level})/2$ for one-sided tests, and $(1 - \text{conf.level})$ for two-sided tests.

`conf.level`: scalar between 0 and 1. The default value is 0.95, in which case the size of the test is 0.05 for two-sided tests and 0.025 for one-sided tests.

`power`: scalar between 0 and 1, specified power for a study. Default is `1-size` for one-sided tests and `1-size/2` otherwise, if neither `alpha` nor `beta` are specified.

Alternately, `power="calculate"`, indicating that the power should be calculated (the sample size and alternative hypothesis should be specified).

`alpha`, `beta`, `epsilon`: these allow for more advanced specifications, e.g. different sizes or power for upper and lower tests, or a continuum between one-sided and two-sided tests. See `seqDesign.testSpecifications`.

*** BOUNDARY SHAPE AND CONSTRAINT ARGUMENTS ***

See `seqDesign.boundaryShape`

for a description of arguments related to specifying boundary shapes and constraints: `design.family`, `P`, `R`, `A`, `G`, `early.stopping`, `exact.constraint`, `minimum.constraint`, and `maximum.constraint`

`design.family`: This may be a character string, one of: "OBF", "Pocock", "Wang", "Symmetric", or "Triangular", specifying a special boundary family found in the literature; see `seqDesign.specialFamilies`. Those families imply certain values for the shape parameters listed below.

Alternately, `design.family` may be a character string or "`seqScale`" object to specify that boundary shape parameters and constraint matrices be interpreted on one of these scales: "X" (sample mean), "S" (partial sum), "Z" (Z value), "E" (error spending – accepts additional parameters, see `seqScale`).

P, R, A: boundary shape parameters. Each may be of length 4 (for the "a-d" boundaries), length 2 (if only two boundaries are used), or length 1 (used for all boundaries).

G: vector of length 4; if supplied it is used for starting values for a numerical search.

early.stopping: character string, one of "null", "alternative", or "both", specifying the hypotheses under which early stopping of the trial is possible. The default is "both" for one-sided tests, and "alternative" otherwise. Specifying certain other arguments causes this to be ignored.

exact.constraint, minimum.constraint, maximum.constraint: either `seqBoundary` objects, or matrices with 4 columns (corresponding to the "a-d" boundaries) and `nbr.analyses` rows (in this case the `design.family` scale is used), or a `seqDesign` object containing the corresponding constraint. Missing values (NA) indicate that a constraint is not active.

display.scale: a character string or `seqScale` object indicating the scale for displaying results; see `seqScale` for choices.

VALUE

a group sequential design object with class "`seqDesign`", and components:

call: the function call used to create the object.

date: date this design was created.

specification: list containing values of the arguments specified in the call.

model: list of class `seqModel`, specifying the probability model, and related quantities

hypothesis: list of class `seqHypothesis`, specifying the test type, power, and related quantities.

parameters: list of class `seqParameters`, specifying the number of analysis times, shape parameters and constraints for boundaries, and related quantities.

boundary: matrix with class `seqBoundary` specifying the boundaries and scale for this design.

std.bounds: matrix with class `seqBoundary` giving the standardized boundaries of the design.

*** DETAILS OF A GROUP SEQUENTIAL DESIGN OBJECT. ***

See `seqDesign.object`, `seqModel.object`, `seqHypothesis.object`,
`seqParameters.object`, `seqBoundary.object`, `seqScale.object` for details regarding the internal structure of a group sequential design object.

SAMPLE SIZE, POWER, ALTERNATIVE, AND BOUNDARIES

To extract any of the key results, use the following:

```
sampleSize(result)
seqExtract(result, "power")
seqExtract(result, "theta.alt")
seqBoundary(result)
seqBoundary(result, scale=lt;see belowgt;)
```

Boundaries (and constraints) are matrices with `nbr.analyses` rows and four columns, labeled "`a`", "`b`", "`c`", and "`d`". "`a`" is the lower boundary; sample results below this boundary at any time cause a decision for the lower hypothesis (e.g. the alternative hypothesis in a one-sided design with `test.type="less"`, or the null hypothesis in a one-sided design with `test.type="greater"`). Results above the upper "`d`" boundary cause a decision for the upper hypothesis. The "`b`" and "`c`" boundaries are only active in two-sided tests, where results between these two boundaries cause a decision for the null hypothesis. The "`a`" or "`b`" boundary may contain infinite values, depending on the value of `early.stopping`.

The boundaries may be displayed (or constraints passed to `seqDesign`) on different scales. Let `X[i]` be the *i*th observation, with mean zero under the null hypothesis; for example, for a one-sample binomial proportions model `X[i] = Y/i - p0`, where `Y` is the number of successes after `i` trials and `p0` is the proportion specified by the null hypothesis. Let `K` be the number of analysis times, and `N[k]` be the number of observations at the `k`th analysis time. Then the `display.scale` and `design.family` arguments correspond to boundaries on the following scales:

```
"partial sum" = S[k] = sum(X[1:N[k]])
"sample mean" = Xbar[k] = S[k] / N[k]
"normalized z-value" = Z[k] = Sk / (sigma sqrt(N[k]))
"fixed sample p-value" = P[k] = pnorm(Z[k])
```

where `sigma` is the standard deviation of a single observation (under the null hypothesis). See `seqScale`

for a description of other scales.

UPDATING A DESIGN

`update` allows you to modify an existing `seqDesign` object, changing one or more arguments. The first argument to `update` should be a `seqDesign` object. Subsequent arguments are the same as for the

`seqDesign` function, and replace arguments used previously. To delete a previous argument, give the argument without a value, e.g.

```
update(object, exact.constraint = )
```

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

See `seqDesign.specialFamilies`

for references for the O'Brien and Fleming ("OBF"), Pocock ("Pocock"), Wang ("Wang"), Whitehead and Stratton ("Triangular"), and Emerson and Fleming ("Symmetric") families.

SEE ALSO

For details on arguments, see: `seqScale`, `seqBoundary`, `seqDesign.probabilityModel`, `seqDesign.multiple.arms`, `seqDesign.testSpecifications`, `seqDesign.boundaryShape`, and `seqDesign.specialFamilies`. To go directly to the details help file containing a particular argument, append the argument name to "seqDesign..", e.g. `seqDesign..prob.model`.

`seqFixDesign` creates a new design with a single analysis time, matching either the power or maximum sample size of an existing design. `update.seqDesign` creates a new design, using the same arguments as an existing design except for specified arguments.

For details about the "seqDesign" object, see: `seqDesign.object`, `seqModel.object`, `seqHypothesis.object`, `seqParameters.object`, `seqBoundary.object`, and `seqScale.object`.

For extracting part of the "seqDesign" object, see: `seqBoundary`, `seqExtract`, `confidenceLevel`, `powerTest`, `sampleSize`.

Functions which use the results include: `seqInference`, `seqOperatingChar`, `confidenceInterval.seqDesign`, `seqPlotASN`, `seqPlotInference`, `seqPlotPower`, `seqPlotStopProb`,

For monitoring group sequential trials, see: `seqMonitor`.

EXAMPLE

```
## create a two-side test with O'Brien-Fleming boundaries
ssr.ofn <- seqDesign(prob.model = "proportions",
                      null.hypothesis = .5, variance = .25, nbr.analyses = 4,
                      sample.size = 100, P = c(1,Inf,Inf,1),
                      test.type = "two.sided", conf.level = .95,
                      display.scale = "S")

## Create a test with a Pocock lower boundary:
ssr.pocn <- update(ssr.ofn, P = c(0.5,Inf,Inf,1))
```

```

## Here is a design with custom boundaries on the Z-scale:
bound <- seqBoundary(rbind(c(-1.645,0,0,Inf),
                           qnorm(c(.0225,.0225,.9775,.9775))),
                      scale = "Z")

protn <- seqDesign(prob.model = "proportions", arms = 2,
                    variance = c(0.2100, 0.1716), nbr.analyses = 2,
                    sample.size = 850, power = 0.9775, test.type = "two.sided",
                    display.scale = "X", exact.constraint = bound)

protn

## convert this to a standard Emerson-Fleming one-sided test
symmn <- update(protn, P = 1, test.type = "greater",
                  exact.constraint=)
summary(symmn)
plot(symmn)
seqPlotPower(symmn)
seqPlotASN(symmn)
seqPlotStopProb(symmn)
seqPlotInference(symmn)

# Create a two-sided design with O'Brien-Fleming boundaries.
des1 <- seqDesign(prob.model = "normal", alt.hypothesis = 1,
                   nbr.analyses = 4, power = .975, design.family = "OBF")
des1

# Create a two-sided design with Pocock boundaries.
des2 <- seqDesign(prob.model = "normal", alt.hypothesis = 1,
                   nbr.analyses = 4, power = .975, design.family = "Pocock")
plot(des2)
des2

# Create a design for a binary response with O'Brien-Fleming
# boundaries using a specified sample size. The default null
# hypothesis is no difference between treatment and control.
des3 <- seqDesign(prob.model = "proportions", nbr.analyses = 4,
                  power = .975, sample.size = 100, design.family = "OBF")
plot(des3)
des3

# Create a 1-arm design for a binary response with Pocock
# boundaries using a specified sample size. The default null
# hypothesis is theta = 0.5.
des4 <- update(des3, arms = 1, design.family="Pocock")
plot(des4)
des4

# Change the previous design to a one-sided hypothesis test.

```

```
# This gives a warning message because the Pocock design family
# is no longer used when test.type = "greater" is specified.
des5 <- update(des4, test.type = "greater")
plot(des5)
des5
```

seqDF.ASN Coerce ASN & quantiles to data frame for plotting

DESCRIPTION

Convert the ASN and sample size quantile curves for group sequential design into a data frame used for plotting with Trellis graphics.

```
seqDF.ASN(x, prob = .75, asn = T, digits = 3)
```

REQUIRED ARGUMENTS

x: a list of group sequential design objects inheriting from "seqDesign" or a list of objects inheriting from "seqOperatingChar". The list must have unique names.

prob: a vector of probabilities between 0 and 1 for which the sample size quantiles should be included in the data frame. If no quantiles are desired set **prob** = NULL.

asn: if TRUE, then the ASN curve will be included in the data frame.

digits: the number of digits to use in forming the name for the sample size quantile curve.

VALUE

a data frame with components:

theta: the values of theta at which the power curve is evaluated.

sample.size: the value of the ASN curve and sample size quantile curve.

quantile: a factor variable indicating which curve: "asn" or one of `paste(round(100*prob,digits), "%-t = "")`

design: a factor variable indicating the name of the corresponding group sequential design.

DETAILS

This function is used by **seqPlotASN** , and is not normally called directly.

SEE ALSO

`seqPlotASN, seqDesign , seqOperatingChar` .

seqDF.Power Coerce power to data frame for plotting

DESCRIPTION

Convert the power curves for group sequential design into a data frame used for plotting with Trellis graphics.

```
seqDF.Power(x, reference = NULL)
```

REQUIRED ARGUMENTS

x: a list of group sequential design objects inheriting from "seqDesign" or a list of objects inheriting from "seqOperatingChar". The list must have unique names.

reference: NULL, or the name of one the components of **x**; in the latter case the corresponding power will be used as a reference, and the data frame will contain the difference between the power curve for other designs and the reference power curve.

VALUE

a data frame with components:

theta: the values of theta at which the power curve is evaluated.

power: the power at theta

hypothesis: a factor variable indicating the alternative hypothesis: "lower" or "upper".

design: a factor variable indicating the name of the corresponding group sequential design.

DETAILS

This function is used by **seqPlotPower**, and is not normally called directly.

SEE ALSO

seqPlotPower, seqDesign.

seqDF.seqBoundary Coerce boundaries to data frame for plotting

DESCRIPTION

Convert the decision boundaries for group sequential design object or boundary object into a data frame for plotting with Trellis graphics.

```
seqDF.seqBoundary(x, labels = NULL)
seqDF.seqDesign(x, display.scale = "X", labels = NULL)
```

REQUIRED ARGUMENTS

x: a list of "seqBoundary" objects, or list of "seqDesign" objects. If **labels** is not supplied then the list must have names, which are unique and non-blank.

display.scale: a character or character string specifying the scale on which to represent the boundaries. Possible values are "S", "X", "Z", "P", "E", "B" and "St" or equivalently, "partial.sum", "sample.mean", "z.value", and "p.value", "error.spend" "bayes" or "standardized".

labels: vector the same length as **x**; if supplied then these labels are used as the names of the designs.

VALUE

a data frame with components:

boundary: the decision boundaries for the group sequential design.

sample.size: the sample size corresponding to the boundaries.

design: a factor variable indicating the name of the corresponding group sequential design, or the label.

DETAILS

These functions are called by **seqPlotBoundary** , and are not normally called directly.

SEE ALSO

seqPlotBoundary, **seqDesign** , **seqBoundary** .

seqDF.seqInference Coerce inference to data frame for plotting

DESCRIPTION

Convert the inferences for group sequential design object into a data frame used for plotting with Trellis graphics.

```
seqDF.seqInference(x, ordering = "a", estimate = "BAM")
```

REQUIRED ARGUMENTS

x: a list of group sequential design objects inheriting from "**seqDesign**" or a list of objects inheriting from "**seqInference**". The list must have unique names.

OPTIONAL ARGUMENTS

ordering: character, either "a" (analysis-time ordering) or "m" (mean ordering); see **seqInference**.

estimate: character, one of "MLE", "MUE", "BAM", "RBadj"; see **seqInference**.

VALUE

a data frame with components:

stats: a numeric vector giving the inferences.

sample.size: a numeric vector giving the sample.size at which the inferences are made.

which: a factor indicating which inference is given by **stats**: one of "observed", "p.value", "estimate", "ci.lower", "ci.upper", "theta.null", "theta.lower", or "theta.upper".

labels: a factor for labeling the points at which the inferences are made. This is used by **seqPlotInference** to group inferences by boundary.

design: a factor variable indicating the name of the corresponding group sequential design.

DETAILS

This function is used by **seqPlotInference**, and is not normally called directly.

SEE ALSO

seqPlotInference, **seqInference**, **seqDesign**.

seqDF.StopProb Coerce stopping probability to data frame for plotting

DESCRIPTION

Convert the stopping probabilities for group sequential design into a data frame used for plotting with Trellis graphics.

```
seqDF.StopProb(x, condition = 0)
```

REQUIRED ARGUMENTS

x: a list of group sequential design objects inheriting from "seqDesign" or a list of objects inheriting from "seqOperatingChar". The list must have unique names.

condition: integer; if positive then **condition** indicates that the stopping probabilities should be conditional upon not stopping at the **condition**th analysis time or earlier.

VALUE

a data frame with components:

theta: the values of theta at which the power curve is evaluated.

time: the stopping time

hypothesis: a factor variable indicating the hypothesis: "lower" , "null", "upper", or "total" where "total" represents the combined stopping probability at a given analysis time.

design: a factor variable indicating the name of the corresponding group sequential design.

prob: the stopping probability (or conditional stopping probability).

cumprob: the cumulative stopping probability (or conditional cumulative probability).

DETAILS

This function is used by **seqPlotStopProb** , and is not normally called directly.

SEE ALSO

seqPlotStopProb, **seqDesign** , **seqOperatingChar** .

seqExtract Extract group sequential design parameters.

DESCRIPTION

Extract parameters from a **seqDesign** object, or from other objects related to group sequential designs.

This function is an S Version 3 generic (see Methods); method functions can be written to handle specific S Version 3 classes of data. Classes which already have methods for this function include:

seqDesign , **seqOperatingChar** , **seqInference** , **seqHypothesis**

```
seqExtract(x, what, ...)
```

REQUIRED ARGUMENTS

x: a group sequential design object of class "**seqDesign**", or certain objects which include or are part of a "**seqDesign**" object.

what: character, indicating what parameter (input value or output quantity) to extract. Current choices are:
"**confidence.level**" (1 minus the Type I error rate),
"**power**" (at the hypotheses specified when creating the design),
"**theta.null**" (natural parameter at the null hypothesis),
"**theta.alt**" (natural parameter at the alternative hypothesis),
"**theta.beta**" (natural parameter used for one-sided power),
"**test.type**" (two-sided, etc.),
"**sample.size**".

These are choices for a **seqDesign** object; other objects may have a more restricted set of choices.

OPTIONAL ARGUMENTS

...: Additional arguments; these are not currently used.

VALUE

The component requested.

DETAILS

The requested parameters can also be extracted in other ways from the object, e.g. the sample size (or vector of analysis times) can be obtained using **x\$parameters\$sample.size**.

The sample size can also be obtained using the **sampleSize** function. Similar single-purpose functions exist for power (**powerTest**) and confidence level (**confidenceLevel**).

SEE ALSO

seqDesign.object , **sampleSize** , **powerTest** , **confidenceLevel** .

EXAMPLE

```
des <- seqDesign(sample.size=300, nbr.analyses=3, P=1)
seqExtract(des, "sample.size")
sampleSize(des)
```

seqFixDesign Create a Fixed Sample Design

DESCRIPTION

Create a fixed-sample hypothesis test which corresponds to a group sequential design.

```
seqFixDesign(x, match.power = T)
```

REQUIRED ARGUMENTS

x: a group sequential design object inheriting from "seqDesign".

OPTIONAL ARGUMENTS

match.power: if TRUE, then the power of the fixed sample design will be matched to the power of the group sequential design. If FALSE, then the sample size for the fixed sample design is set to the maximum sample size for the group sequential design.

VALUE

a "seqDesign" object with one analysis time. See **seqDesign.object** for information on the structure of this object.

DETAILS

This is implemented by calling **update**, which in turn calls **seqDesign**. Any constraints are deleted.

If the group sequential design **x** is between a one-sided and a two-sided test, then the design created is a two-sided fixed test. Hybrid designs of this form correspond to when either **epsilon[1]** or **epsilon[2]** is not equal to zero or one.

SEE ALSO

seqDesign, **seqDesign.object** , **update.seqDesign** .

EXAMPLE

```
symm <- seqDesign(sample.size = 500, nbr.analyses = 4, P = .5)
fxd1 <- seqFixDesign(symm)
fxd2 <- seqFixDesign(symm, match.power = F)
sampleSize(symm)
sampleSize(fxd2)
plot(symm, fxd1, fxd2, fixed = F)
plot(symm, fxd1, fxd2, fixed = F, superpose.design = T)
# In the last plot the right-most times are jittered.
```

seqGetKeyParameters Get Trellis Parameters for Key

DESCRIPTION

Sets the Trellis parameters for displaying a key (legend) in plots for group sequential tests.

```
seqGetKeyParameters(obj, which, type = "superpose.line")
```

REQUIRED ARGUMENTS

obj: a Trellis graphics object.

which: the component names of the Trellis parameters, or numerical indices.

OPTIONAL ARGUMENTS

type: the Trellis parameters to set.

VALUE

the Trellis parameters **type** with the components named by **which** replaced from **obj**.

DETAILS

This is called by various plot methods for group sequential designs, and is not intended for general use.

SEE ALSO

trellis.par.get (located in S-PLUS Help).

EXAMPLE

```
des.tr <- plot(seqDesign(nbr.analyses = 4,
                         sample.size = 200, null.hypothesis = 100))
seqGetKeyParameters(des.tr, which = c(2,4,6),
                    type = "superpose.symbol")
```

seqHypothesis.object Hypotheses Object

DESCRIPTION

Objects which inherit from **seqHypothesis**,

GENERATION

Objects which inherit from "**seqHypothesis**" are returned as the **hypothesis** component of a **seqDesign** object created by the **seqDesign** function. There is no **seqHypothesis** function.

METHODS

The class "**seqHypothesis**" has **print** and **seqExtract** methods.

VALUE

The "**seqHypothesis**" object is a list with the following components:

- test.type** character[length 1]: a character string specifying the alternative type. Possible types are "greater", "less", "two.sided", "equivalence", and "advanced".
- size**: scalar level of significance (Type I error).
- power**: scalar, specified power for a study, or power at the design alternative hypothesis.
- alpha**: vector of length 2, giving the level of significance (Type I error) for the upper and lower hypothesis tests.
- beta**: vector of length 2 giving the power of the test at the lower and upper alternative hypotheses.
- theta.null**: a number giving the value of the natural parameter theta at the null hypothesis (difference in parameter values for a 2-arm test).
- theta.alt**: scalar, natural parameter at the design alternative hypothesis.
- theta.alpha**: a vector of length 2 giving natural parameters corresponding to the upper and lower null hypotheses.
- theta.beta**: a vector of length 2 giving natural parameters corresponding to one-sided power calculations.
- prob.model**: character, indicating the probability model. Possibilities are: "N", "L", "B", "O", "P", and "H", which correspond to "normal", "lognormal", "proportions", "odds", "poisson", and "hazard", respectively.
- log.transform**: logical, if TRUE then the summary measure of outcome is reported on a logarithmic scale, e.g. log odds ratio, log hazard ratio.

arms: a number, giving the number of arms in the design. The value 0 corresponds to the continuous regression case.

SEE ALSO

`seqDesign, seqDesign.object`.

seqInference.object Inference Object

DESCRIPTION

Objects which inherit from **seqInference**.

GENERATION

Objects which inherit from the class "**seqInference**" are returned from the **seqInference** function. This is used by the **summary**, and **seqPlotInference** functions.

METHODS

The class "**seqInference**" has methods for **print**, **plot**, **[**, and **seqExtract**.

VALUE

A "**seqInference**" object is a data frame of class **c("seqInference", "data.frame")**, with variables:

analysis.index: indices of analysis times.

observed: observed statistics.

sample.size: a vector of the sample sizes at the analysis times.

MLE: maximum likelihood estimates (based on the observed statistics).

MUE.meanOrder, **MUE.timeOrder**: median-unbiased estimate, calculated using mean ordering and analysis time ordering, respectively.

BAM: bias-adjusted mean estimate.

RBadj: Rao-Blackwell adjusted estimate; under some conditions this is a UMVUE (uniformly minimum-variance unbiased estimate).

PvalueL.meanOrder, **PvalueU.meanOrder**: p-values for lower and upper hypotheses tests, using mean ordering. Only one of these is present for a one-sided test.

PvalueL.timeOrder, **PvalueU.timeOrder**: p-values for lower and upper hypotheses tests, using analysis time ordering. Only one of these is present for a one-sided test.

CILo.meanOrder, **CIHi.meanOrder**: endpoints of a two-sided confidence interval with confidence **conf.level**, using mean ordering.

CILo.timeOrder, **CIHi.timeOrder**: endpoints of a two-sided confidence interval with confidence **conf.level**, using analysis time ordering.

labels: labels for printing and plotting (need not be present).

In addition, "**seqInference**" objects have the following attributes:

scale: the scale on which the observed values are measured.

conf.level: the confidence level of the confidence intervals.

design: the original design used to compute the inferences.

ordering: a character indicating how to compute the confidence intervals: "m" corresponds to sample mean ordering, "a" corresponds to analysis time ordering, and "b" to both.

at.the.boundary: if TRUE, the the inferences were computed at the boundaries of the design.

NOTE

Use `print.data.frame(x)` to use normal data frame printing.

SEE ALSO

`seqDesign`, `seqInference` , `seqPlotInference`

seqInference Inference for a Hypothesis Test

DESCRIPTION

Compute point estimates, p-values, and confidence intervals.

```
seqInference(x, analysis.index = <>see below>,
             observed = <>see below>,
             inScale = attr(x$boundary, "scale"),
             ordering = "b",
             lower = x$hypothesis$test.type != "greater",
             upper = x$hypothesis$test.type != "less",
             conf.level = 95/100,
             labels = NULL)
```

REQUIRED ARGUMENTS

x: a group sequential design object inheriting from "**seqDesign**", or a list of such objects.

OPTIONAL ARGUMENTS

analysis.index: a vector of analysis times at which to make inferences. By default, these are the analysis times of the boundaries of **x**.

observed: a vector of observed statistics, the same length as **analysis.index**. By default, **observed** are the boundaries of **x**.

inScale: the scale for the observed values; this is ignored if no observed values are supplied.

ordering: a character indicating how to compute the confidence intervals, p-values, and median-unbiased estimates: "**m**" corresponds to sample mean ordering (Emerson and Fleming, 1990), "**a**" corresponds to analysis time ordering (Tsiatis, Rosner, and Mehta, 1984), and "**b**" to both.

lower: if TRUE, then p-value(s) are computed for the lower alternative.

upper: if TRUE, then p-value(s) are computed for the upper alternative.

conf.level: the confidence level for two-sided confidence intervals based on the observed values.

labels: character vector the same length as **analysis.index**, giving the names of the data pairs (**analysis.index**, **observed**). This is used for printing and plotting.

VALUE

If `x` is a "seqDesign" object, then the function returns a "seqInference" object. Otherwise, `x` should be a list of "seqDesign" objects, and the function returns a list of "seqInference" objects, of class "seqInferenceList".

A "seqInference" object is a data frame of class `c("seqInference", "data.frame")` with variables: `analysis.index`, `observed`, `sample.size`, `MLE`, `MUE.meanOrder`, `MUE.timeOrder`, `BAM`, `RBadj`, `PvalueL.meanOrder`, `PvalueU.meanOrder`, `PvalueL.timeOrder`, `PvalueU.timeOrder`, `CIlo.meanOrder`, `CIhi.meanOrder`, `CIlo.timeOrder`, `CIhi.timeOrder`, and optionally `labels`. The object also possesses attributes. See `seqInference.object` for details.

DETAILS

There is a special print method; use `print.data.frame` to use normal data frame printing.

The (`analysis.time`, `observed`) pairs should not be in the continuation region at any time – they should be values for which the trial stops. Missing values (NA) will be returned for any pairs which are in the continuation region.

The analysis time ordering is not defined for group sequential designs with `a[k] lt; b[k] lt; c[k] lt; d[k]`. See Emerson (2000) for details.

When both `upper` and `lower` are true, all one-sided p-values are returned, but the `print.seqInference` function reports an overall p-value which is twice the smaller of the one-sided p-values.

Confidence intervals are always two-sided.

Inferences are not calculated for a multiple-arms trial (`arms gt; 2`). In this case the function returns a character string indicating that inferences are not calculated.

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

Emerson, S. and Fleming, T. R. (1990). Parameter estimation following group sequential hypothesis testing. *Biometrika*, Vol. 77, 875-892.

Tsiatis, A. A., Rosner, G. L. and Mehta, C. R. (1984). Exact confidence intervals following a group sequential test. *Biometrics*, Vol. 40, 797-803.

Emerson, S. and Kittelson, J.M. (1997). A computationally simpler algorithm for the UMVUE of a normal mean following a group sequential trial. *Biometrics*, Vol. 53, 365-369.

SEE ALSO

```
seqDesign, summary.seqDesign, seqInference.object, seqPlotInference  
, seqDF.seqInference .
```

EXAMPLE

```
des <- seqDesign(prob.model = "proportions",  
                  sample.size = c(90, 130, 180),  
                  conf.level = .95, P = 1, null.hypothesis = 0.25)  
seqInference(des)  
seqInference(des, c(1,2),  
             observed = c(12/45 - 14/45, 19/65 - 25/65))
```

seqModel.object Sequential Design Model Object

DESCRIPTION

Objects which inherit from **seqModel**,

GENERATION

Objects which inherit from the class "**seqModel**" are created by the **seqDesign** function, and returned as the **model** component in a "**seqDesign**" object.

METHODS

The class "**seqModel**" has a **print** method.

VALUE

The "**seqModel**" object is a list with components:

prob.model: a character string specifying which model family: "normal", "lognormal", "proportions", "odds", "poisson", or "hazard".

log.transform: logical, if TRUE then the summary measure of outcome is reported on a logarithmic scale, e.g. log odds ratio, log hazard ratio.

arms: an integer value equal to the number of arms (0 indicates the continuous regression case).

ratio: if **arms**>0, a vector of length **arms**, giving the relative number of observations assigned to each arm.

null.hypothesis: the parameters (e.g. normal mean) under the null hypothesis.
If **arms**=2, a vector of length 2, giving the parameter for each arm under the null hypothesis.

alt.hypothesis: vector of parameters for the design alternative hypothesis. If **arms**=2, a vector of length 2, giving the parameter for each arm under the alternative hypothesis; if the second arm is a control group then **alt.hypothesis**[2]=**null.hypothesis**[2].

variance: character, indicating how variance is calculated, or numeric, indicating the value of the variance of a single observation. If **arms**=2, a numerical value is a vector of length 2, giving the variance for each arm.

theta.null: the natural parameter (e.g. binomial proportion, normal mean) under the null hypothesis; if **arms**=2, the difference in parameter values between the arms.

sigma.sqr: a measure of the variance of estimates, incorporating the **ratio** information in the two-arms case.

psi: a constant, used together with a link function for transforming between mu (mean) and theta; see Emerson (2000).

DETAILS

The object is actually created in the `seqDesignCtoS` function, which is called by `seqDesign`.

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

SEE ALSO

`seqDesign`, `seqDesign.object`

seqMonitor Group Sequential Monitoring

DESCRIPTION

Modify a group sequential design for monitoring a clinical trial.

```
seqMonitor(x, response,
           treatment = rep(1, length(response)),
           future.analyses,
           constraint.scale = x$monitor$constraint.scale,
           min.increment = 0.1, maxiter = 20, variance)
update(seqMonitor object, <>new arguments>>)
```

REQUIRED ARGUMENTS

- x:** a group sequential monitoring object inheriting from "seqMonitor", or, if the call to this function represents the first interim analysis, a group sequential design object inheriting from "seqDesign".
- response:** measures of response for each patient appropriate to the probability model specified in **x**. This will be a vector for all probability models except for proportional hazards, for which it should be a Surv object.
- treatment:** a vector of indicators of treatment group. This must agree with the number of arms specified in **x**: either unspecified or a constant vector when **arms** is 1, a vector with values of 0 (for the comparison group) or 1 (for the treatment group) when **arms** is 2, or a numeric vector with at least two distinct values when **arms** is 0 (a regression model).

OPTIONAL ARGUMENTS

future.analyses: a vector specifying future analysis times. If the last value of this vector is greater than 1, then this is taken to mean that the maximal sample size of the study is constrained at that value. Otherwise (if the last value is missing, or no value exceeds 1) this argument gives relative spacings, and the maximal sample size is adjusted to maintain the power specified in **x** for the alternative specified in **x**. By default this is taken from the seqMonitor object, or from the estimated analysis times in the seqDesign object if this is the first monitoring analysis, and timings are relative. **future.analyses** may be NULL or a length-0 vector to indicate that no future analyses should be performed.

Missing values are replaced with sample sizes evenly spaced between the adjacent nonmissing values. A particularly useful option is to specify **future.analyses=c(NA,NA,...,1)**

to request equally-spaced analysis times between the current number of observations (or events, for the proportional hazards model) and the maximal number chosen to maintain power.

constraint.scale: A character specifying the scale on which the boundaries at previously conducted analyses should be constrained. Choices include "X" (sample mean scale), "Z" (Z statistic scale), "P" (fixed sample P value scale), and "E" (error spending scale). The choice of "E" corresponds to the Lan-DeMets type I error spending approach when the maximal sample size is constrained, and to the Pampallona, Tsiatis, and Kim type I and II error spending approach when the maximal sample size is unconstrained. Only "E" is permissible when the original group sequential design was defined using an error spending family. All choices are permissible for other design families.

min.increment: An indicator of the minimal relative increment in sample size for the next interim analysis following the current analysis. This plays two roles. First, if any previously-planned analysis time approximately matches the sample size represented in **response**, this analysis is assumed to be that analysis, and the actual time replaces the planned time; the tolerance for determining this is **min.increment** (default 10%) times the previously-planned maximum sample size. Second, as the maximum sample size is adjusted to maintain power, additional analysis times may be deleted so that no interval is less than a **min.increment** fraction of the maximum sample size. The last analysis time will never be removed (an error occurs instead).

maxiter: Maximum number of iterations to allow for convergence. If the function fails to converge the current best estimate is returned.

variance: numeric value of variance. By default this is calculated from **treatment**. It may be supplied for the "normal", "lognormal", and "proportions" probability models. See **seqDesign..variance** for details, except that here **variance** must be numeric, not character.

VALUE

a group sequential monitoring object with class "**seqMonitor**", which inherits from "**seqDesign**". It has components:

call: the function call used to create the object.

date: date this monitoring object was created.

specification: list containing values of the arguments specified in the call.

design: list of class **seqDesign**, containing the original group sequential design object.

seqDesignCall: the function call to seqDesign used to create the original design.

statistics: list containing the results of the current and all prior analyses of the data. Elements of the list include:

N – a vector of sample sizes at which the analyses occurred;

variance – a list of variance estimates at each of the analyses;
X, **Z**, and **P** – vectors of test statistics on the corresponding scales which were computed at each of the analyses;
response – a list of the response measurements at each of the analyses;
treatment – a list of the treatment vectors at each of the analyses;
ratio – a list of the ratio of treatment assignments at each of the analyses.

monitor: list containing information about the monitoring history for the clinical trial. Elements of the list include:

constraint.scale – a character indicating the constraint scale used in computing the monitoring boundaries;

future.analyses – a vector of the estimated timing of future analyses;

Xbnd, **Zbnd**, and **Ebnd** – matrices of the exact stopping boundaries used at this and previous analyses.

model: list of class **seqModel**, specifying the probability model, and related quantities.

hypothesis: list of class **seqHypothesis**, specifying the test type, power, and related quantities.

parameters: list of class **seqParameters**, specifying the number of analysis times, shape parameters and constraints for boundaries, and related quantities.

boundary: matrix with class **seqBoundary** specifying the boundaries and scale for the monitoring bounds.

std.bounds: matrix with class **seqBoundary** giving the standardized boundaries.

*** DETAILS OF A GROUP SEQUENTIAL MONITORING OBJECT. ***

See **seqDesign.object** , **seqModel.object** , **seqHypothesis.object** , **seqParameters.object** , **seqBoundary.object** , and **seqScale.object**

for details regarding the internal structure of a group sequential monitoring object. All components except **seqDesignCall**, **design**, **statistics**, and **monitor** are present in a **seqDesign** object.

UPDATING A DESIGN

update (**update.seqDesign**) modifies a **seqMonitor** object. The first argument to **update** should be a **seqMonitor** object. Subsequent arguments are the same as for the **seqMonitor** function, and replace arguments used previously. To delete a previous argument, give the argument without a value, e.g. **update(object, exact.constraint =)**.

DETAILS

The process of conducting a clinical trial can be thought of as proceeding through three stages: design, monitoring, and analysis. At the design stage, `seqDesign()` allows a user to select a general stopping rule that has the desired operating characteristics. At that stage, however, the number and timing of analyses are usually only estimates of when the interim analyses will actually occur. If the actual schedule of analyses varies from those estimates, slight modifications will need to be made to the stopping boundaries in order to obtain the stopping rule appropriate to the actual number and timing of analyses. `seqMonitor()` is the function that computes those modified boundaries in a way such that the type I statistical error is maintained at the level specified during the design stage. In addition to maintaining the type I error, a user can decide which one of the maximal sample size or the type II statistical error (or equivalently, the power of the study) should be maintained.

If the maximal sample size is not constrained at some fixed value, `seqMonitor` will choose the maximal sample size to maintain the estimated statistical power to detect the design alternative as specified at the design stage. This estimate of the maximal sample size will be based on a schedule of monitoring which reflects all prior interim analyses and an estimate of the number and timing of future analyses. `seqMonitor` rounds all analysis times upward, so that the actual power may be slightly greater than originally specified. However, no rounding is performed if the maximal analysis time is constrained and at least one value of `future.analyses` is non-integer.

By default, the relative spacings of future analysis times are taken from those estimated at a prior interim analysis or during the design stage.

At each analysis, `seqMonitor()` performs an analysis of the clinical trial data accrued to date according to the probability model specified in the original design. Then using the information regarding the number and timing of all previous interim analyses along with the sample size at the current analysis, the stopping boundary appropriate for use at this analysis is calculated. That stopping boundary is computed according to the design family parameters specified at the design stage constrained to agree with the boundaries actually used at all prior interim analyses. Owing to the need to use estimates of the variability of the test statistic, there can be slight variations in the way that those constraints behave according to the scale used for the constraints. Hence the user may specify the constraint scale. If the original design family was the error spending family, the boundaries must be constrained on the error spending scale. However, if the original design family was any other choice, the boundaries can be constrained on any of the valid monitoring scales: sample mean,

Z statistic, fixed sample P value, or error spending scale. For this latter choice, the stopping rule specified at the design stage is converted to the error spending scale and then used as the basis for an error spending function, with interpolation used to define the error spending function at analysis times not specified at the design stage.

REFERENCES

- Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)
- Lan, K.K.G. and DeMets, D.L. (1983). Discrete sequential boundaries for clinical trials, *Biometrika*, Vol 70, 659-663.
- O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials, *Biometrics*, Vol 35, 549-556.
- Pampallona, S.A., Tsiatis, A.A., and Kim K.M. (1995). Spending functions for the type I and type II error probabilities of group sequential tests. Technical Report, Department of Biostatistics, Harvard School of Public Health

SEE ALSO

`seqDesign` (for creating a design object, and note that functions listed there that operate on a design object can also operate on a monitoring object), `update.seqDesign` (for updating a design or monitoring object).

See `seqDesign.object` , `seqModel.object` , `seqHypothesis.object` , `seqParameters.object` , `seqBoundary.object` , and `seqScale.object` for details regarding the internal structure of a group sequential monitoring object.

For additional examples see Case Study 2 – `case2.seqtrial` .

EXAMPLE

```
# Create a one-sided group sequential test of a greater
# hypothesis with O'Brien-Fleming efficacy boundary and
# less conservative futility boundary:
orig.design <- seqDesign(prob.model = "proportions",
  null.hypothesis = .5, variance = "alternative", nbr.analyses = 5,
  alt.hypothesis = .75, P = c(0.75, 1))

# First interim analysis constraining on sample mean scale
# (obs.resp1 and trtmnt1 are vectors containing
# the binary response and the treatment assignment 0 or 1,
# respectively. Simulated data under the alternative
# hypothesis are provided here):
trtmnt1 <- rep(0:1, c(40,35)); set.seed(0)
obs.resp1 <- c(rbinom(40, 1, .5), rbinom(35, 1, .75))
interim1 <- seqMonitor(orig.design, obs.resp1, trtmnt1)
```

```

print(interim1)

# Examine operating characteristics of modified boundaries:
plot(interim1, original=T)
seqPlotPower(orig.design, interim1)
seqPlotASN(interim1)
seqPlotStopProb(interim1)

# Examine how monitoring boundaries might differ if maximum
# sample size is constrained to sample size estimated at
# design stage:
interim1.maxN <- update(interim1,
                        future.analyses=sampleSize(orig.design))
plot(orig.design, interim1, interim1.maxN, superpose.design=T)
seqPlotPower(orig.design, interim1, interim1.maxN)

# Examine how monitoring might differ with equally-spaced
# future analyses
interim1.equal <- update(interim1, future.analyses = c(NA, NA, 1))
sampleSize(interim1)
sampleSize(interim1.equal)

# Examine how monitoring might differ if this analysis were last:
interim1.final <- update(interim1, future.analyses = NULL)
seqPlotPower(orig.design, interim1, interim1.maxN, interim1.final)

# Examine how first interim analysis would differ if
# Lan-DeMets were used relative to constrained maximal
# sample size with sample mean constraints:
interim1.LD <- update(interim1, constraint.scale="E",
                      future.analyses = sampleSize(orig.design))
plot(orig.design, interim1.maxN, interim1.LD, superpose.design=T)
seqPlotPower(orig.design, interim1.maxN, interim1.LD)

# Examine how first interim analysis would differ if
# Pampallona, Tsiatis, and Kim were used:
interim1.PTK <- update(interim1, constraint.scale="E")
plot(orig.design, interim1, interim1.PTK, superpose.design=T)
seqPlotPower(orig.design, interim1, interim1.PTK)

# Second interim analysis constraining on sample mean scale
# (obs.resp2 and trtmnt2 are vectors containing the binary
# response and the treatment assignment 0 or 1, respectively).
# Simulated data under the alternative hypothesis are used here):
trtmnt2 <- c(trtmnt1, rep(0:1, c(45,50))); set.seed(1)
obs.resp2 <- c(obs.resp1, rbinom(45, 1, .5), rbinom(50, 1, .75))
interim2 <- seqMonitor(interim1, obs.resp2, trtmnt2)
plot(interim2)
plot(orig.design, interim1, interim2, superpose.design=T)
seqPlotASN(orig.design, interim1, interim2)

```

```
# Sample size if only one more analysis to be done, and want to  
# maintain power specified at design stage:  
max(sampleSize(update(interim2, future.analyses=1)))
```

seqOperatingChar.object Operating Characteristics Object

DESCRIPTION

Objects which inherit from **seqOperatingChar**.

GENERATION

Objects which of class "seqOperatingChar" are created by the **seqOperatingChar** function. This is used by the **summary.seqDesign** , **seqPlotASN**, **seqPlotPower**, and **seqPlotStopProb** functions.

METHODS

The class "seqOperatingChar" has methods for **print**, **plot** , **[**, and **seqExtract**.

VALUE

A "seqOperatingChar" is a data frame with variables:

theta: the values of the parameter **theta** at which the operating characteristics are computed.

power.lower: the probability that a trial is stopped by hitting the lower ("a") boundary,

$$\Pr(S[M] \leq a[M] \mid \theta),$$

where M is the time a trial stops and S[M] is the statistic at that time.

power.upper: the probability that a trial is stopped by hitting the upper ("d") boundary,

$$\Pr(S[M] \geq d[M] \mid \theta).$$

asn: the average (expected) sample size,

$$\sum(N[k] \Pr(M = k \mid \theta); k = 1:K)$$

where N[k] is the sample size at the kth analysis time, and K is the total number of analysis times.

lower.stop.prob: matrix with K columns, giving the probability of stopping for the lower alternative at times 1:K.

null.stop.prob: matrix with K columns. For a two-sided design this gives the probabilities of stopping for the null alternative. For a one-sided design, this is identically zero, and stopping probabilities for the null hypothesis are found in **lower.stop.prob** (if **test.type="greater"**) or **upper.stop.prob** (if **test.type="less"**).

upper.stop.prob: matrix with K columns, giving the probability of stopping for the upper alternative at times 1:K.

stop.prob: sum of the previous three.

The data frame also has the following attributes:

K: the number of analysis times

sample.size: the sample sizes of the analysis times

alternative.type: the alternative type of the design

design: the `seqDesign` object the operating characteristics were computed from.

NOTE

The data frame contains matrices giving stopping probabilities. To obtain the whole matrix, use e.g. `result$lower.stop.prob`. To extract a single column, use e.g. `result$lower.stop.prob[,2]`.

NOTE

The print method prints only part of the object, and does so in a way that differs from normal data frame printing. Use `print.data.frame(x)` to use normal data frame printing.

SEE ALSO

`seqDesign`, `seqOperatingChar`, `seqPlotASN`, `seqPlotPower`, `seqPlotStopProb`

seqOperatingChar Operating Characteristics

DESCRIPTION

Computes the upper and lower power, ASN, and stopping probabilities for group sequential designs.

```
seqOperatingChar(x, theta = <<see below>>, how.many = 50,
                 range.theta = <<see below>>, power = c(.01,.99),
                 upper = <<see below>>, lower = <<see below>>)
seqOC( <<same as seqOperatingChar, but quicker to type>> )
```

REQUIRED ARGUMENTS

x: a group sequential design object inheriting from "seqDesign", or a list of such objects.

OPTIONAL ARGUMENTS

theta: a vector giving the values of theta at which the operating characteristics are to be computed.

how.many: desired length of **theta** (used if **theta** is not supplied).

range.theta: vector of length two; if this is supplied and **theta** is not, then **theta** is created as a sequence of **how.many** equally-spaced values over this range.

power: vector of values between 0 and 1 (exclusive). If neither **theta** nor **theta.range** is supplied, then **theta** is set to those values that achieve these values of **power**. In the special case that **power** is of length 2, it is interpreted as a range and expanded to a sequence of length **how.many** values over that range.

upper: if TRUE, then the upper power is used when solving values of **theta** that match the specified **power**. By default this is true if the test type is "two-sided" or "greater".

lower: if TRUE, then the lower power is used when solving values of **theta** that match the specified **power**. By default this is true if the test type is "two-sided" or "less".

VALUE

If **x** is a "seqDesign" object, then the function returns an object of class "seqOperatingChar". Otherwise, if **x** is a list of "seqDesign" objects, the function returns a list of "seqOperatingChar" objects which use a common **theta**; the list has class "seqOperatingCharList".

A "seqOperatingChar" object is a data frame with **length(theta)** rows and variables **theta**, **power.lower**, **power.upper**, **asn**, **lower.stop.prob**

, `null.stop.prob` , `upper.stop.prob` , `stop.prob` (total). The latter four are matrices. For a one-sided design `null.stop.prob` is identically zero, and stopping probabilities for the null hypothesis are found in one of `lower.stop.prob` or `upper.stop.prob`. The data frame also has attributes. For details see `seqOperatingChar.object` .

DETAILS

There is a special print method; use `print.data.frame` to use normal data frame printing.

If both `upper` and `lower` are TRUE, then both are used, and `theta` has length equal to `2*how.many`.

If `x` is a list of `seqOperatingChar` objects created by `seqOperatingChar`, then `upper` is TRUE if it would be for any of the objects, and similarly for `lower`.

SEE ALSO

`seqDesign`, `summary.seqDesign` , `seqOperatingChar.object` , `seqPlotASN` , `seqPlotPower` , `seqPlotStopProb` , `seqDF.ASN` , `seqDF.Power` , `seqDF.StopProb` .

EXAMPLE

```
des <- seqDesign(sample.size = c(60,90,120), P = .5)
seqOC(des)
## Compute operating characteristics for five values of
## theta spanning the lower and upper alternative hypothesis;
## this is the default behaviour of the summary function
theta.null <- seqExtract(des, "theta.null")
theta.min <- min(seqExtract(des, "theta.alt"))
theta.max <- max(seqExtract(des, "theta.alt"))
seqOperatingChar(des, theta = c(theta.min,
  (theta.min+theta.null)/2, theta.null,
  (theta.max+theta.null)/2, theta.max))
```

seqParameters.object Parameters for Group Sequential Design Object

DESCRIPTION

Objects which inherit from **seqParameters**.

GENERATION

This object is created by **seqDesign**, and is the **parameters** component of a **seqDesign** object.

METHODS

The class "**seqParameters**" has methods for the **print** generic function.

VALUE

The "**seqParameters**" object is a list with components:

nbr.analyses: scalar, number of analysis times

sample.size: vector of length **nbr.analyses**, containing the number of observations at each of the analysis times.

alpha: vector of length 2, giving the level of significance (Type I error) for the upper and lower hypothesis tests.

beta: vector of length 2, giving the

epsilon: vector of length 2, representing the type of test. **c(1,0)** corresponds to a test type of "less", **c(1,0)** to "greater", **c(1,1)** to "two-sided". The elements of **epsilon** should be between 0 and 1 with **sum(epsilon) >= 1**. The lower hypothesis is specified by **epsilon[1]** and the upper hypothesis is specified by **epsilon[2]**.

design.family: object of class "**seqScale**", indicating the scale on which results should be displayed by default.

A, P, R, G: each vectors of length 4, containing shape parameters for boundaries "a", "b", "c", "d".

exact.constraint, minimum.constraint, maximum.constraint: objects of class "**seqBoundary**" – matrices with **nbr.analyses** rows and 4 columns, corresponding to boundaries "a", "b", "c", "d", plus attributes indication the scale the constraints are measured on. These give exact, minimum, and maximum constraints on the decision boundaries. A missing value **NA** in any position indicates the lack of a corresponding constraint.

SEE ALSO

seqDesign, seqDesign.object, seqBoundary.object, seqScale.object

seqPHSampleSize Proportional Hazards Sample Size (older version)

DESCRIPTION

Compute sample size needed to observe desired number of events in a proportional hazards survival model.

```
seqPHSampleSize(nevents, hazardRatio, controlMedian,  
accrualTime, followupTime, ratio=c(1,1))
```

NOTE

This function is deprecated; the use of **seqPHSubjects** is recommended instead.

OPTIONAL ARGUMENTS

nevents: total number of events needed for proportional hazards clinical trial design as might be calculated by **seqDesign(prob.model="hazard",...)**.

hazardRatio: the hazard ratio comparing the treatment arm to the comparison arm.

controlMedian: the median time to event on the comparison arm.

accrualTime: the time period during which subjects will be accrued to the study. It is assumed that the accrual rate is uniform during this time period.

followupTime: the additional time that subjects will be followed after the accrual period has ended. (Note that the total study time will be **accrualTime + followupTime**.)

ratio: the ratio of randomization of subjects to the treatment and comparison arms. This can be a vector of length two or a scalar, in the latter case the second element is presumed to be 1.

VALUE

A vector of sample sizes representing the number of subjects which should be accrued to attain the desired number of events.

DETAILS

If all arguments (except **ratio**) are scalar the result is a scalar.

In proportional hazards models, the statistical information about the treatment effect is roughly proportional to the number of events observed, rather than the number of subjects on study. Therefore, **seqDesign()** computes the number of events needed to attain the desired power. For trial planning purposes, it is necessary to estimate

the number of subjects required to observe that number of events in a reasonable period of time.

In order to calculate the number of subjects needed, some assumptions need to be made about the underlying distribution of times to event and times to censoring. A commonly used model is to assume that censoring occurs only due to staggered study entry and, furthermore, that study accrual occurs according to a uniform distribution during the accrual period. Under this model, subjects can be followed for events for some additional follow-up time following the accrual of the last subject to the study. The distribution of times to event is assumed to be exponential.

REFERENCES

O'Brien, P. C. and Fleming, T. R. (1979). A multiple testing procedure for clinical trials, *Biometrics*, Vol 35, 549-556.

SEE ALSO

`seqPHSubjects`.

EXAMPLE

```
# One-sided symmetric test of a lesser hypothesis
# with O'Brien-Fleming boundary shape function:
obf <- seqDesign(prob.model = "hazard", nbr.analyses = 5,
                   test.type = "less", alt.hypothesis = 0.75)

# explore sample sizes required under null and alternative
# hypotheses, various accrual and follow-up times, and selected
# possibilities for median time to event on comparison arm
HR <- rep(c(1, 0.75), each=9)
CtrlMdn <- rep(rep(2:4, each=3), 2)
AccrTm <- rep(c(4,4,5,4,4,5,4,4,5), 2)
FUTime <- rep(c(1,2,1,1,2,1,1,2,1), 2)
N <- seqPHSampleSize(nevents = max(sampleSize(obf)),
                      hazardRatio = HR, controlMedian = CtrlMdn,
                      accrualTime = AccrTm, followupTime = FUTime)
cbind(HR, CtrlMdn, AccrTm, FUTime, N)

n <- seqPHSubjects(obf, controlMedian = 2:4,
                    accrualTime = 4:5, followupTime = 1:2)
n
summary(n)
```

seqPHSubjects Proportional Hazards Sample Size

DESCRIPTION

Compute sample size needed to observe desired number of events in a proportional hazards survival model.

```
seqPHSubjects(design, controlMedian,
               accrualTime, followupTime, rate,
               nEvents, hazardRatio, ratio,
               lambda0 = log(2)/controlMedian)
```

REQUIRED ARGUMENTS

design: a `seqDesign` object created by `seqDesign(prob.model="hazard", ...)`, or corresponding `seqMonitor` object. This is required unless all of `nEvents`, `hazardRatio`, and `ratio` are supplied.

controlMedian: numeric, median survival time for the control arm. This is ignored (and not required) if `lambda0` is supplied.

OPTIONAL ARGUMENTS Supply any two of `accrualTime`, `followupTime`, and `rate`; the third is calculated.

accrualTime: numeric, accrual time.

followupTime: numeric, followup-up time after accrual is completed.

rate: numeric, rate at which patients accrue.

nEvents: total number of events needed for proportional hazards clinical trial design; default is `sampleSize(design)`.

hazardRatio: the hazard ratio comparing the treatment arm to the comparison arm; default is extracted from `design`.

ratio: the ratio of randomization of subjects to the treatment and comparison arms. This can be a vector of length two or a scalar. By default this is extracted from `design`.

lambda0: event (death) rate for the control arm, parameter of the exponential distribution. This may be supplied instead of `controlMedian`.

VALUE

a data frame with class `seqPHSubjects`, containing components `accrualTime`, `followupTime`, `rate`, `hazardRatio`, `lambda0` or `controlMedian`, `nSubjects`, and 'analysisTimes'. There is one row in the data frame for every combination of the input variables; for example if argument `accrualTime` contains three candidate accrual times, `followupTime`

contains four values, and `controlMedian` contains 2 values, the data frame has 24 rows. All arguments except `design` and `ratio` may be vectors (if `ratio` is a vector of length 2 it is converted to a scalar). The `accrualTime`, `followupTime`, or `rate` component is calculated based on the other values in the row.

`nSubjects`: is the total number of subjects enrolled, `rate*accrualTime`.

`analysisTimes`: is a matrix with one column for each (interim and final) analysis time (the `nbr.analyses` argument to `seqDesign`), giving the estimated time at which the analysis should be performed to obtain the number of events specified by `seqDesign` for each analysis time. This component is not present if `design` was not supplied, or if there was only one analysis time.

The result also has attributes

`compute`: one of "accrualTime", "followupTime", or "rate", indicating what was calculated (the others were input arguments).

`out.attrs`: a list with components `dim` and `dimnames`, suitable for turning the calculated result into an array with dimension names corresponding to the input variables.

DETAILS

The model is that patients accrue to the study uniformly at rate `rate` from time 0 to `accrualTime`; the study continues after accrual stops for an additional time `followupTime`. Survival time for individual patients follows an exponential distribution with hazard rate `lambda0` for the control group (calculated from `controlMedian`), and hazard rate `lambda0 * hazardRatio` for the treatment group.

If `followupTime` is calculated, then `followupTime=Inf` (infinity) indicates that the `accrualTime` and `rate` were insufficient to provide the desired number of events, and a negative value of `followupTime` indicates that the desired number of events was obtained that long before the end of accrual.

When calculating `followupTime` or `accrualTime`, and when calculating `analysisTimes`, the function uses `uniroot` for numerical root finding.

There is a `summary` method for this class, which converts the calculated result into an array. There is also a subscripting method, which returns an ordinary data frame.

SEE ALSO

`summary.seqPHSubjects`, `seqDesign` , `expand.grid` .

EXAMPLE

```

hdes1 <- seqDesign(prob.model="hazard",
                     alt.hypothesis = .8, ratio = c(1,1),
                     nbr.analyses = 4, test.type="less")

# Calculate accrual time
seqPHSubjects(hdes1, 5, followupTime = 20, rate = 43)

# Calculate followup time, one vector input
seqPHSubjects(hdes1, 5, accrualTime = 30, rate = c(43:44))

# Calculate rate,
seqPHSubjects(hdes1, 5, accrualTime = 30:31, followupTime = 20:21)

# Options for printing, results as array or ordinary data frame
temp <- seqPHSubjects(hdes1, 5, accrualTime = 30:31,
                       followupTime = 20:21)
temp # print as a data frame
temp2 <- summary(temp) # results as arrays
temp2 # print calculated rate as an array
temp2$analysisTimes # calculated analysis times, interim and final
aperm(temp2$rate, c(3,1,2)) # reorder the dimensions

# Plotting
temp <- seqPHSubjects(hdes1, 25, accrualTime = 30, followupTime = 20:30)
xyplot(rate ~ followupTime | factor(hazardRatio),
       type="l", data=temp)
xyplot(rate ~ followupTime, subset = (hazardRatio == 1),
       type="l", data=temp)
xyplot(rate ~ followupTime, groups = hazardRatio,
       panel = panel.superpose,
       type="l", data=temp)
xyplot(nSubjects ~ followupTime, groups = hazardRatio,
       panel = panel.superpose,
       type="l", data=temp)

```

seqPlotASN Plot ASN and Sample Size Quantile Curves

DESCRIPTION

Plot the ASN curve (expected sample size) and the sample size quantile curves for a group sequential design.

```
seqPlotASN(x, ..., subset = T, fixed = T,
           prob = .75, asn = T,
           superpose.design = <<see below>>,
           superpose.quantiles = !superpose.design,
           add.key = SeqTrial.options()$add.key,
           key.args = list(space = "top"),
           frame = sys.nframe())
```

REQUIRED ARGUMENTS

x: a `seqDesign` object, a list of `seqDesign` objects, a `seqOperatingChar` object, or a list of `seqOperatingChar` objects.

OPTIONAL ARGUMENTS

....: Additional `seqDesign` or `seqOperatingChar` objects (`x` and these arguments must all be the same type).

Graphical parameters and arguments to high-level graphics functions may also be supplied as arguments to this function (see `par`, `trellis.args`, `plot.default`, and `title`).

subset: an expression to plot only a subset of the stopping rules. If TRUE, then everything is plotted.

fixed: if TRUE then a fixed design (single analysis time) with the same power as `x` is created and also plotted.

prob: a vector of probabilities between 0 and 1 for which the sample size quantiles should be displayed. If no quantiles are desired, then you should set `prob = NULL`.

asn: if TRUE, then the ASN curve is displayed.

superpose.design: if TRUE, then for each ASN curve and sample size quantile curve, all of the designs are superposed on the same panel. By default, `superpose.design = TRUE` if more than one design is displayed (e.g. if `fixed = TRUE`).

superpose.quantile: if TRUE, then for each design, the ASN curve and sample size quantile curves are superposed on the same panel.

add.key: if TRUE and `superpose.design == TRUE` or `superpose.quantiles == TRUE`, a key is added to the plot identifying the different designs or quantiles.

key.args: additional arguments can be passed to the trellis `key` function. This can be used to control the placement of the key. By default, the key is placed on top of the plot.

frame: the frame in which to evaluate the `subset` argument.

VALUE

an object of class `trellis` which is automatically plotted by `print.trellis`.

DETAILS

The ASN curve gives the average number of observations until stopping, as a function of `theta`:

```
ASN(theta) = sum(N[k] Pr(M = k | theta); k = 1:K)
```

where K is the number of analysis times, `N[k]` is the number of observations at the kth analysis time, and M is the (random) stopping time for the trial.

The sample size quantile curve is the analysis time `Q(p ; theta) = N[k]` such that

```
Pr(M <= k | theta) >= p
```

and

```
Pr(M >= k | theta) >= 1-p.
```

This function converts the design objects to a data frame using `seqDF.ASN` and constructs a trellis expression. Other named arguments can be supplied to the plot. These arguments are passed through ... to trellis graphics: see `trellis.args` for details. Customized displays can be obtained by working directly with trellis graphics; see `seqDF.ASN`.

Fixed designs are created by `seqFixDesign`, which determines what alternative hypothesis is used when matching power.

Quantile curves are step functions, but may not appear that way in these plots if the plotting points are too widely spaced.

SEE ALSO

`seqDesign`, `seqDF.ASN` , `seqFixDesign` . The following are located in S-PLUS Help: `par`, `trellis.args`, `plot.default`, and `title`.

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses=3, P=1)
asym1 <- update(symm, P = c(.9,1))
asym2 <- update(symm, P = c(.7,1))
#This plots symm, seqFixDesign(symm), asym1, and asym2:
seqPlotASN(symm, asym1, asym2, prob = NULL)
#Plots the sample size quantile curve for prob = .5 and .9
seqPlotASN(symm, prob = c(.5,.9), asn = F)
```

seqPlotBoundary Plot a Group Sequential Design

DESCRIPTION

Plot the decision boundaries for a group sequential design.

```
plot(x, ..., fixed = <<see below>>, superpose.design = <<see below>>,
      observed = NULL, N = NULL,
      display.scale, yaxis.expand = 1.1,
      ylim = <<see below>>, xlim = <<see below>>,
      vertical.lines = <<see below>>
      lwd = 3, connect.lwd = 1,
      add.key = SeqTrial.options()$add.key,
      key.args = list(space = "top"),
      subset = T, frame = sys.nframe())
plot.seqDesign( <<same as above>> )
plot.seqBoundary( <<same as above>> )
seqPlotBoundary( <<same as above>> )
plot.seqMonitor( <<same as above>>, original=T )
```

REQUIRED ARGUMENTS

x: a **seqDesign** object, a list of **seqDesign** objects, a **seqBoundary** object, a list of **seqBoundary** objects, or (for **plot.seqMonitor**) a **seqMonitor** object.

OPTIONAL ARGUMENTS

...: Additional **seqDesign** or **seqBoundary** objects (**x** and these arguments must all be the same type).

Graphical parameters and arguments to high-level graphics functions may also be supplied as arguments to this function; (see **par**, **trellis.args**, **plot.default**, and **title**).

All subsequent arguments must be given by name (not position) and may not be abbreviated.

fixed: logical, if TRUE and **x** is a **seqDesign** object (or list thereof), then a fixed design (single analysis time) with the same power as **x** (or **x[[1]]**) is created and the corresponding boundary also plotted. Default is TRUE if only one **seqDesign** object is supplied.

superpose.design: if TRUE, then the boundaries for all designs are superposed on a single panel. Default is TRUE if there are exactly two designs.

observed: NULL, or a vector of observed statistics on the same scale as the boundaries.

plot.seqMonitor ignores the **observed** and **N** arguments, and extracts this information from the **seqMonitor** object.

N: vector the same length as **observed**; the sample size corresponding to **observed**.

display.scale: the scale on which to plot the boundaries. This may be a character string, or a **seqScale** object; see **seqScale**. Possible values are "S", "X", "Z", "P", "E", "B", "St", "C", or "H", or equivalently, "partial.sum", "sample.mean", "z.value", "p.value", "error.spend", "bayesian", "standardized", "conditional", or "predictive". Some of these require or allow arguments such as parameters of a prior distribution; to supply those pass a **seqScale** object rather than a character string. This is ignored by **plot.seqMonitor**, which always uses "X".

yaxis.expand: the y-axis is expanded by a factor of **yaxis.expand** beyond the range of the boundaries.

ylim: by default, **ylim** is the range of the boundaries, expanded by the factor **yaxis.expand**.

xlim: by default, **xlim** is **range(0,unlist(xvalues))** where **xvalues** are all the x-values to be plotted.

vertical.lines: if TRUE, vertical lines are drawn connecting the stopping boundaries to the edges of the plotting region. By default vertical lines are drawn except when plotting on error spending, conditional futility, or predictive futility scales.

lwd: integer giving line width for vertical lines; default 3.

connect.lwd: integer giving line width for connecting lines; default 1, except the default is 3 for error spending, conditional futility, or predictive futility scales.

lty, **connect.lty**, **col**, **connect.col**: graphical parameters for line type and color, for vertical and connecting lines (these are ... arguments).

add.key: if TRUE and **superpose.design = TRUE**, a key is added to the plot identifying the different designs.

key.args: additional arguments can be passed to the trellis **key** function. This can be used to control the placement of the key. By default, the key is placed on top of the plot.

subset: an expression to plot only a subset of the stopping rules. If TRUE, then everything is plotted.

frame: the frame in which to evaluate the **subset** argument.

original: logical, if TRUE then the original design is plotted.

VALUE

an object of class **trellis** which is automatically plotted by **print.trellis**.

DETAILS

`plot.seqDesign`, `plot.seqBoundary`, and `plot.seqMonitor` call `seqPlotBoundary`.

This function converts the design objects to a data frame using `seqDF.seqDesign` and constructs a trellis expression using the panel functions `panel.seqBoundary` and `panel.seqBoundary.superpose`; the latter is used if `superpose.design = TRUE`.

Other graphical arguments can be supplied to the plot. These arguments get passed through ... to trellis graphics. See `trellis.args`, `panel.seqBoundary.superpose`, and `panel.seqBoundary`

for details. Customized displays can be obtained by working directly with trellis graphics: see `seqDF.seqDesign`.

In the case of `seqMonitor`, arguments `observed` and `N` are taken from the observed data and analysis times contained in the `seqMonitor` object, and argument `display.scale` is always "X"; any values supplied for these three arguments are ignored.

SEE ALSO

`seqDesign`, `seqScale`, `seqMonitor`, `seqDF.seqDesign`, `panel.seqBoundary`, `panel.seqBoundary.superpose`, `trellis.args` (located in S-PLUS Help).

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P = 1)
asym1 <- update(symm, P = c(.9,1))
asym2 <- update(symm, P = c(.7,1))
#This plots symm, seqFixDesign(symm), asym1, and asym2
plot(symm, asym1, asym2)
plot(symm, asym1, asym2, superpose.design = T)

# Supply main title as a list to make it larger; see trellis.args
plot(symm, main = list("Main title", cex=2.5))
```

seqPlotInference Plot a Group Sequential Design

DESCRIPTION

Plot the inferences for a group sequential design object.

```
seqPlotInference(x, ...,
                  ordering, estimate = "BAM",
                  subset = T,
                  xlim = <<see below>>, ylim = <<see below>>,
                  yaxis.expand = 1.1, superpose = F,
                  add.key = SeqTrial.options()$add.key,
                  key.args = list(space = "top"),
                  frame = sys.nframe())
plot.seqInference(x, <<same as above>>)
```

REQUIRED ARGUMENTS

x: a **seqDesign** object, a list of **seqDesign** objects, a **seqInference** object, or a list of **seqInference** objects.

OPTIONAL ARGUMENTS

....: Additional **seqDesign** or **seqInference** objects (**x** and these arguments must all be the same type).

Graphical parameters and arguments to high-level graphics functions may also be supplied as arguments to this function (see **par**, **trellis.args**, **plot.default**, and **title**).

ordering: character, either "a" (analysis-time ordering) or "m" (mean ordering); determines which ordering will be used for p-values, confidence intervals, and the MUE estimate. The default is taken from the "**ordering**" attribute of the first "**seqInference**" object, otherwise "m".

estimate: character string, determines which estimate is plotted, one of "BAM" (bias-adjusted mean), "MLE" (maximum-likelihood), "MUE" (median-unbiased), "RBadj" (Rao-Blackwell adjusted).

subset: an expression to plot only a subset of the stopping rules. If TRUE, then everything is plotted.

xlim: by default, **xlim** is **range**(0, 1.1***unlist**(**xvalues**)) where **xvalues** are all the x-values to be plotted.

ylim: by default, **ylim** is the range of values y values to be plotted, extended by the factor **yaxis.expand**.

yaxis.expand: scalar, factor by which to extend the y axis (used if **ylim** is missing).

superpose: if TRUE, then all the inferences for a design will be superposed on a single panel. By default, if more than one design is supplied, then **superpose** = FALSE.

add.key: if TRUE and **superpose** = TRUE, a key will be added to the plot identifying the different designs.

key.args: additional arguments can be passed to the trellis **key** function. This can be used to control the placement of the key. By default, the key is placed on top of the plot.

frame: the frame in which to evaluate the **subset** argument.

VALUE

an object of class **trellis** which is automatically plotted by **print.trellis**.

DETAILS

This function converts the design objects to a data frame using **seqDF.seqInference** and constructs a trellis expression using the panel functions **panel.seqInference**.

Other graphical arguments can be supplied to the plot. These arguments get passed through ... to trellis graphics. See **trellis.args** and **panel.seqInference**

for details. Customized displays can be obtained by working directly with trellis graphics: see **seqDF.seqInference** .

SEE ALSO

seqDesign, **seqDF.seqInference** , **panel.seqInference** , **summary.seqDesign** , The following are located in S-PLUS Help: **par**, **trellis.args**, **plot.default**, and **title**.

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P = 1)
# This produces a tabular version of the inferences
seqInference(symm)
# This produces a graphical display
seqPlotInference(symm)
# Again, but with specified observed values:
inf.symm <- seqInference(symm, analysis.index = 2, observed = .35)
inf.symm
plot(inf.symm)
```

seqPlotPower Plot a Power Curve

DESCRIPTION

Plot the power curve for a group sequential design.

```
seqPlotPower(x, ..., subset = T, reference = F,
             fixed = F, match.power = F,
             yrange = 0.01, superpose.design = <<see below>>,
             superpose.hypothesis = F,
             add.key = SeqTrial.options()$add.key,
             key.args = list(space = "top"),
             frame = sys.nframe())
```

REQUIRED ARGUMENTS

x: a `seqDesign` object, a list of `seqDesign` objects, a `seqOperatingChar` object, or a list of `seqOperatingChar` objects.

OPTIONAL ARGUMENTS

...: Additional `seqDesign` or `seqOperatingChar` objects (**x** and these arguments must all be the same type).

Graphical parameters and arguments to high-level graphics functions may also be supplied as arguments to this function (see `par`, `trellis.args`, `plot.default`, and `title`).

subset: an expression to plot only a subset of the stopping rules. If TRUE, then everything is plotted.

reference: if TRUE or if **reference** inherits from "seqDesign", then a fixed design (or **reference**) is used as a reference design. In this case, the difference between each power curve and the reference power curve is displayed: `beta[x](theta) - beta[ref] (theta)`.

fixed: logical, if TRUE or if **reference == TRUE**, then then a fixed design (single analysis time) with the same power as **x** (or **x[[1]]**) is created and the corresponding power curve also plotted.

match.power: logical, if TRUE then the fixed design matches power, otherwise uses the same maximum sample size as **x** (or **x[[1]]** if **x** is a list of designs).

yrange: if **reference = TRUE** or if **reference** inherits from "seqDesign", then **ylim** is extended if necessary to include **yrange**; if `length(range) == 1` then **ylim** extends to at least `c(-yrange, yrange)` (this is useful for producing plots with a common vertical scale).

superpose.design: if TRUE, then the power curves for all designs are superposed on the same panel. Default is TRUE if there are two or more designs and **superpose.hypothesis=FALSE**.

superpose.hypothesis: if TRUE, then the lower and upper power curves are superposed on the same panel.

add.key: if TRUE and either **superpose** argument is TRUE, a key is added to the plot identifying the different designs or hypotheses.

key.args: additional arguments can be passed to the trellis **key** function. This can be used to control the placement of the key. By default, the key is placed on top of the plot.

frame: the frame in which to evaluate the **subset** argument.

VALUE

an object of class **trellis** which is automatically plotted by **print.trellis**.

DETAILS

This function converts the design objects to a data frame using **seqDF.Power** and constructs a trellis expression. Other graphical arguments can be supplied to the plot. These arguments are passed through ... to trellis graphics: see **trellis.args** for details. Customized displays can be obtained by working directly with trellis graphics; see **seqDF.Power**

Fixed designs have a single analysis time, and are created by **seqFixDesign**.

SEE ALSO

seqDesign, **seqDF.Power**, **seqFixDesign**. The following are located in S-PLUS Help: par, trellis.args, plot.default, and title.

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P = 1)
asym1 <- update(symm, P = c(.9,1))
asym2 <- update(symm, P = c(.7,1))
#This plots symm, seqFixDesign(symm), asym1, and asym2:
seqPlotPower(symm, asym1, asym2, superpose.design = T)
seqPlotASN(symm, asym1, asym2, prob = NULL)
# Now use symm as a reference
seqPlotPower(asym1, asym2, fixed = F, reference = symm,
            superpose.design = T)
```

seqPlotStopProb Plot Stopping Probabilities

DESCRIPTION

Plot the stopping probabilities for a group sequential design.

```
seqPlotStopProb(x, ..., subset = T, cumulative = T,
                 condition = FALSE,
                 which = "all", ylim = c(0,1),
                 superpose.design = F,
                 superpose.hypothesis = !superpose.design,
                 superpose.time = T,
                 add.key = SeqTrial.options()$add.key,
                 key.args = list(space = "top"),
                 frame = sys.nframe())
```

REQUIRED ARGUMENTS

x: a `seqDesign` object, a list of `seqDesign` objects, a `seqOperatingChar` object, or a list of `seqOperatingChar` objects.

OPTIONAL ARGUMENTS

....: Additional `seqDesign` or `seqOperatingChar` objects (**x** and these arguments must all be the same type).

Graphical parameters and arguments to high-level graphics functions may also be supplied as arguments to this function (see `par`, `trellis.args`, `plot.default`, and `title`).

subset: an expression to plot only a subset of the stopping rules. If TRUE, then everything is plotted.

cumulative: if TRUE, the cumulative probability of stopping is displayed. If `superpose.hypothesis == FALSE`, then the cumulative probability is accumulated separately for each hypothesis; otherwise, the cumulative probabilities are accumulated across hypotheses.

condition: either FALSE or a positive integer representing an analysis time; in the latter case probabilities plotted are conditional on not stopping at that analysis time or earlier.

which: a character string indicating which hypotheses are to be displayed: "total", "lower", "null", "upper", or "all".

ylim: the range of the y-axis.

superpose.design: if TRUE, then the stopping probabilities all of the designs are superposed on the same panel.

superpose.hypothesis: if TRUE, then the stopping probabilities for all hypotheses are superposed on the same panel.

superpose.time: if TRUE, then the stopping probabilities at all analysis times are superposed on the same panel.

add.key: if TRUE and **superpose.design** == TRUE or **superpose.hypothesis** == TRUE, a key is added to the plot identifying the designs or hypotheses.

key.args: additional arguments can be passed to the trellis **key** function. This can be used to control the placement of the key. By default, the key is placed on top of the plot.

frame: the frame in which to evaluate the **subset** argument.

VALUE

an object of class **trellis** which is automatically plotted by **print.trellis**.

DETAILS

This function converts the design objects to a data frame using **seqDF.StopProb** and constructs a trellis expression. If **superpose.hypothesis** = TRUE, then the special panel function **panel.seqPlotStopProb** is used. Otherwise, the panel function is **panel.xyplot** or **panel.superpose**.

Other graphical arguments can be supplied to the plot. These arguments are passed through ... to trellis graphics: see **trellis.args** and **panel.seqPlotStopProb**

for details.

Some graphical arguments must be vectors of certain lengths. For example, if specifying colors, the basic rule is that **length(col)** should match the number of levels being superposed, with some exceptions. If **superpose.hypothesis**=TRUE, then **length(col)**=4 (there are 4 hypothesis, including "total"); **superpose.time** is irrelevant. If **superpose.design**=TRUE, then number of designs; except if only 1 design, and **superpose.time**=TRUE, may match number of analysis times. If **superpose.time**=TRUE and no other superpose, either 1 or number of analysis times.

Customized displays can be obtained by working directly with trellis graphics; see **seqDF.StopProb**

to create the necessary data frame.

The default values for **cumulative** is FALSE if **which** == "all" and **superpose.hypotheses** == FALSE.

SEE ALSO

seqDesign, **seqDF.StopProb** , **panel.seqPlotStopProb** . The following are located in S-PLUS Help: **par**, **trellis.args**, **plot.default**, and **title**.

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P=1)
seqPlotStopProb(symm)
seqPlotStopProb(symm, superpose.hypothesis = F,
                layout = c(2,2))
seqPlotStopProb(symm, superpose.time = F)
asym1 <- update(symm, P = c(.9,1))
asym2 <- update(symm, P = c(.7,1))
seqPlotStopProb(symm, asym1, asym2, which = "total")
seqPlotStopProb(symm, asym1, asym2, which = "total",
                superpose.design = T)
```

seqSampleSizeQuantile Sample Size Quantile Curve

DESCRIPTION

Computes the sample size quantile curve for a group sequential design.

```
seqSampleSizeQuantile(x, prob = c(.5, .75, .9), digits = 4)
```

REQUIRED ARGUMENTS

x: an object of class "seqOperatingChar".

OPTIONAL ARGUMENTS

prob: a numeric vector with values between 0 and 1 specifying the quantiles to compute.

digits: the number of digits to use in labeling the quantiles.

VALUE

A matrix with the sample size quantiles.

DETAILS

This function is used by **seqPlotASN**.

SEE ALSO

seqPlotASN, seqDF.ASN .

seqScale.object Scale Group Sequential Design Object

DESCRIPTION

Objects which inherit from **seqScale**.

GENERATION

This object is created by **seqScale**, and is the **parameters** component of a **seqDesign** object.

METHODS

The class "seqScale" has methods for the **print** generic function.

VALUE

The "seqScale" object is a character string, with the attribute

parameters: a vector of length between 0 and 4, which contains any parameters necessary to define the family.

DETAILS

Possible values for the character string are "S", "X", "Z", "P", "E", "B", "St", "C", "H", which correspond to the "partial.sum", "sample.mean", "z.value", "p.value", "error.spend", "bayesian", "standardized", "conditional", and "predictive" scales, respectively.

Let $X[i]$ be the i th observation (with mean zero under the null hypothesis), K be the number of analysis times, and $N[k]$ be the number of observations at the k th analysis time. Then the **scale** argument corresponds to boundaries on the following scales:

"partial sum" = $S[k] = \text{sum}(X[1:N[k]])$

"sample mean" = $Xbar[k] = S[k] / N[k]$

"z.value" = $Z[k] = (S[k] - \theta_0 N[k]) / (\sigma \sqrt(N[k]))$

"p.value" = $P[k] = \text{pnorm}(Z[k])$ (fixed-sample p-value)

where **sigma** is the standard deviation of a single observation (under the null hypothesis).

SEE ALSO

seqDesign, **seqDesign.object** , **seqScale** .

seqScale Create a seqScale object

DESCRIPTION

Create a scale object which determines the scale for measuring boundaries and constraints in group sequential designs.

```
seqScale(scaleType = "X", threshold, hypTheta = "design",
         priorTheta, priorVariation, pessimism,
         boundaryNumber, scaleParameters = NULL, ...)
```

OPTIONAL ARGUMENTS

scaleType: character string, one of "S", "X", "Z", "P", "E", "B", "St", "C", or "H", Alternately, this may be one of "partial.sum", "sample.mean", "z.value", "p.value", "error.spend", "bayesian", "standardized", "conditional", or "predictive"; each element in the latter set is equivalent to the corresponding element in the first set.

threshold: scalar. Optional for "B" and "H" scales, supply only if **priorTheta** is supplied. For "C" scale, supply iff **hypTheta** is numeric.

hypTheta: length 1. For "C" scale, may be numeric; then **threshold** must be supplied too. For "C" scale, may be one of "design" and "estimate". For "E" scale, numeric, must be supplied iff **boundaryNumber** is.

priorTheta: scalar, optional for "B" and "H" scales.

priorVariation: positive number. Required for "B" and "H" scales.

pessimism: scalar, optional for "B" and "H" (exactly one of **priorTheta** and **pessimism** should be supplied).

boundaryNumber: character, one of "a", "b", "c", or "d", indicating which boundary is to be used in the error spending ("E") scale; must be supplied iff **hypTheta** is.

scaleParameters: vector of parameters; if supplied other parameters except **scaleType** are ignored. See the Technical Overview or the **seqScale** function if you really want to use this.

...: additional arguments, which are passed to **checkSeqScale**, to determine whether a scale is legal for a specified use.

VALUE

An object of class "**seqScale**" – a character string indicating which scale is used, and a attribute **parameters** of length between 0 and 4, containing parameters for that scale.

No parameters are used for the "S", "X", "Z", "P", and "ST" scales.

DETAILS

See the Technical Overview.

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

SEE ALSO

`checkSeqScale`, `seqBoundary` , `seqBoundary.object` , `seqDesign` , `seqDesign.object` , `seqScale.object` .

EXAMPLE

```
seqScale("X")
# seqScale("B") # fails - need to specify Bayesian prior parameters.

# Possible uses for scales:
# * Creating designs -- seqDesign(..., design.family=?)
# * Constraints -- seqDesign(..., exact.constraint=?)
# * Displaying results -- seqDesign(..., display.scale=?)
#                         changeSeqScale(..., outscale=?)
#                         seqBoundary(..., scale=?)
# * Densities -- dSeq(..., inScale=?), pSeq(), qSeq()
# * Inference -- seqInference(..., inScale=?)
# * Monitoring -- seqMonitor(constraint.scale=?)
#
# The list below uses the following codes:
#   D = scale may be used for creating designs and monitoring
#   C = scale may be used for constraints
#   d = scale may be used for display (all may be used for display)
#   f = scale may be used for densities and inference

# Possibilities for parameters with scales:
# (x = real number, p = positive number, l = one of "a","b","c","d")
x <- rnorm(1); x2 <- rnorm(1); p <- rexp(1); l <- "a"
seqScale("X") # DCdf
seqScale("S") # DCdf
seqScale("Z") # DCdf
seqScale("P") # Cdf
seqScale("E") # DCd (use for constraints iff used for design)
seqScale("St") # d
seqScale("E", boundaryNumber = l, hypTheta = x) # d
seqScale("B", priorVariation = p, priorTheta = x, threshold = x2) # df
seqScale("B", priorVariation = p, priorTheta = x) # Cd
seqScale("B", priorVariation = p, pessimism = x) # Cd
seqScale("H", priorVariation = p, priorTheta = x, threshold = x2) # df
seqScale("H", priorVariation = p, priorTheta = x) # Cd
seqScale("H", priorVariation = p, pessimism = x) # Cd
seqScale("C", hypTheta = x, threshold = x2) # df
seqScale("C", hypTheta = "design") # Cd
```

```
seqScale("C", hypTheta = "estimate") # Cd  
# Notes:  
# "St" works for display.scale and seqBoundary(), but not changeSeqScale().  
  
# All scales with parameters can also be specified using the  
# scaleParameters argument - see the Technical Overview.
```

SeqTrial.options Set options for SeqTrial module

DESCRIPTION

Set global variables for the SeqTrial module (group sequential design and monitoring)

```
SeqTrial.options(...)  
SeqTrial.options(add.key = T, plot.ylim = 1.5)
```

OPTIONAL ARGUMENTS

...: a list may be given as the only argument, or a vector of character strings given as the only argument, or any number of arguments may be in the **name = value** form , or no argument at all may be given. See the VALUE and SIDE EFFECTS sections for explanation.

add.key: if TRUE, then the default for all plotting functions is to add a key.

plot.ylim: a parameter which controls clipping on the y-axis; this is not currently used.

DETAILS

The options are saved in a dataset called ".SeqTrial.Options" on the session frame (frame 0). This disappears when you quit S-PLUS. To override the system defaults and save your values for the next session, create a dataset ".SeqTrial.Options" on your working directory.

Other global variables which can affect the behaviour of functions in the SeqTrial module can be set using **options** or **trellis.par.set**.

VALUE

SeqTrial.options always returns a list - even if of length 1. If no argument is given, the list of the current values of all options is returned.

If a character vector is given as the only argument, a list of the current values of the options named by the character vector is returned.

If an object of mode list is given as the only argument, its components will be copied in as the values for options with the corresponding names (the side effect) and a list of the previous values of those options are returned. Generally, the list given as an argument is the value of a previous call to "SeqTrial.options".

Otherwise, arguments must be given in the **name = value** form, and a list is returned of the previous values of the options with these names.

SEE ALSO

The following are located in S-PLUS Help: options, trellis.par.set.

EXAMPLE

```
symm <- seqDesign(sample.size = 300, nbr.analyses = 3, P = 1)
# Turn off the key for all plots
seqopt.old <- SeqTrial.options(add.key = F)
plot(symm)
# restore original options
SeqTrial.options(seqopt.old)
```

summary.seqDesign.object Group Sequential Design Summary Object

DESCRIPTION

Objects which inherit from **summary.seqDesign**,

GENERATION

Objects which inherit from the class "**summary.seqDesign**" are returned from the **summary.seqDesign** function. This is typically called by calling **summary** where the first argument is a **seqDesign** object.

METHODS

The class "**summary.seqDesign**" has **plot** and **print** methods.

VALUE

The "**summary.seqDesign**" object is a list with components:

theta: the values of **theta** at which operating characteristics were computed.

operating.char: an object of class **seqOperatingChar**, containing operating characteristics. This contains power curves, ASN (average sample number), and stopping probabilities.

inference: an object of class **seqInference**, containing inferences (p-values, confidence intervals, etc.) at the boundaries of the design.

design: the original group sequential design object, of class **seqDesign**.

fixed: a group sequential design object with one stopping time, with the same power as **design**.

fixed.operating.char: an object of class **seqOperatingChar**, containing operating characteristics for the fixed design.

The **fixed** and **fixed.operating.char** components are optional.

SEE ALSO

summary.seqDesign, **seqOperatingChar.object** , **seqInference.object** , **seqDesign.object** .

summary.seqDesign Summary of Group Sequential Design

DESCRIPTION

Computes the power, average sample size, stopping probabilities and inferences at the boundaries.

```
summary.seqDesign(object, theta = <<see below>>,
                  fixed = F, ordering = "m",
                  how.many = 5, range.theta = NULL)
```

REQUIRED ARGUMENTS

object: a group sequential design object inheriting from "**seqDesign**".

OPTIONAL ARGUMENTS

theta: a vector giving the values of **theta** at which the operating characteristics are to be computed. By default, **theta** is a vector of length 5 including the values of **theta** at the null and alternative hypotheses.

fixed: logical, if TRUE then a fixed design (single analysis time) is created and its operating characteristics also computed.

ordering: a character indicating what method is preferred for computing inferences – "m" corresponds to sample mean ordering (Emerson and Fleming, 1990) and "a" corresponds to analysis time ordering (Tsiatis, Rosner, and Mehta, 1984), and "b" signifies both. Both are computed in either case, this argument determines which are printed by default.

how.many: integer, desired length of theta (used if theta is not supplied).

range.theta: vector of length two; if this is supplied and **theta** is not, then **theta** is created as a sequence of **how.many** equally-spaced values over this range (a geometric sequence is used in some cases, e.g. for an odds-ratio or hazard ratio).

VALUE

An object of class "**summary.seqDesign**", which is a list containing components that describe operating characteristics (power, average sample size, stopping probabilities) and inferences (estimates, p-values, and confidence intervals). Printing a summary object with the **print** method produces a formatted table of the operating characteristics at **theta** including ASN, lower and upper power, and the sample size and power for the corresponding fixed sample test. Also printed is a formatted table of inferences at the boundaries.

Plotting a summary object produces a set of four plots: a plot of the boundaries (**seqPlotBoundary**), a plot of the power curves (**seqPlotPower**), a plot of the average sample size (**seqPlotASN**), and a plot of the stopping probabilities (**seqPlotStopProb**). Better formatting and resolution can be achieved by evaluating these plots individually.

DETAILS

The analysis time ordering is not defined for group sequential designs with $a[k] < t; b[k] < t; c[k] < t; d[k] \dots$. See Emerson (2000) for details.

See **seqOperatingChar**

and **seqInference**

for details on operating characteristics and inferences for a hypothesis test.

REFERENCES

Emerson, S. (2000). *S+SEQTRIAL: Technical Overview* Research Report No. 98, Insightful Corp., 1700 Westlake Ave N., Seattle, WA, 98109. (S+SeqTrial includes this as an on-line manual.)

Emerson, S. and Fleming, T. R. (1990). Parameter estimation following group sequential hypothesis testing. *Biometrika*, Vol. 77, 875-892.

Tsiatis, A. A., Rosner, G. L. and Mehta, C. R. (1984). Exact confidence intervals following a group sequential test. *Biometrics*, Vol. 40, 797-803.

SEE ALSO

seqDesign, **seqInference**, **seqInference.object**, **seqOperatingChar**, **seqOperatingChar.object**, **plot.seqDesign**, **seqPlotASN**, **seqPlotPower**, **seqPlotStopProb**.

EXAMPLE

```
des <- seqDesign(sample.size = c(100,150),
                  test.type = "greater")
desSummary <- summary(des)
desSummary
print.default(desSummary)
print.default(desSummary$operating.char)
print.default(desSummary$inference)
```

summary.seqPHSubjects Convert seqPHSubjects results to arrays

DESCRIPTION

Convert a **seqPHSubjects** data frame into array(s)

```
summary.seqPHSubjects(object)
print.summary.seqPHSubjects(x, ..., analysisTimes = F)
```

REQUIRED ARGUMENTS

object: a **seqPHSubjects** object.

x: a **summary.seqPHSubjects** object.

OPTIONAL ARGUMENTS

....: optional arguments such as **digits** which used when printing arrays.

analysisTimes: logical flag, if TRUE then an array of analysis times (interim and final) is printed.

VALUE

list with class **summary.seqPHSubjects**, with components:

?: the first component is an array, named "analysisTime", "followupTime", or "rate", depending what was calculated by **seqPHSubjects**. There is one dimension for every vector-valued input argument to **seqDesign**, with dimension names indicating the value of the input. If no inputs were vector-valued this is a scalar.

analysisTimes: this component is present if there is more than one analysis time. This has the same dimensions and names as the first component, plus an additional dimension corresponding to analysis times.

(others): any scalar input arguments are represented as additional list components, each of which has the name of the argument and the corresponding scalar value.

DETAILS

The primary use of this function is for printing the results from **seqPHSubjects** in a different form. The arrays may also be extracted and permuted as desired.

SEE ALSO

seqPHSubjects, **seqDesign**

EXAMPLE

```

hdes1 <- seqDesign(prob.model="hazard",
                     alt.hypothesis = .8, ratio = c(1,1),
                     nbr.analyses = 4, test.type="less")

temp <- seqPHSubjects(hdes1, 5, accrualTime = 30:31,
                      followupTime = 20:21)
temp2 <- summary(temp)
temp2                                # nice printing
print(temp2, digits = 4) # round to four significant places

# Extract results, permute arrays
names(temp2)                         # "rate", "analysisTimes", "controlMedian"
temp2$rate                            # The calculated rate, as an array
aperm(temp2$rate, c(3,1,2)) # reorder the dimensions

```

tutorial.seqtrial Commands for Running the S+SeqTrial Tutorial

DESCRIPTION

The script shown below contains the command-line functions for running the S+SeqTrial tutorial, as described in the *S+SeqTrial User's Guide*.

EXAMPLE

```
# DESIGNING A FIXED SAMPLE TEST
tutorial.fix <- seqDesign(prob.model = "proportions", arms = 2,
                           log.transform = F, size = .025, power = .975,
                           null.hypothesis = c(.6,.6), alt.hypothesis = c(.4, .6),
                           test.type = "less", variance = "alternative")
print(tutorial.fix)
update(tutorial.fix, display.scale = "Z")

seqPlotPower(tutorial.fix)

# DESIGNING A GROUP SEQUENTIAL TEST
tutorial.obf <- update(tutorial.fix, nbr.analyses = 5)
print(tutorial.obf)

seqPlotASN(tutorial.obf, fixed = F)
seqPlotPower(tutorial.obf, tutorial.fix, superpose.design = T)
seqPlotPower(tutorial.obf, reference = tutorial.fix)

tutorial.poc <- update(tutorial.obf, P = .5)
print(tutorial.poc)

seqPlotBoundary(tutorial.poc,tutorial.obf, superpose.design = F)
seqPlotASN(tutorial.poc, tutorial.obf, fixed = F)

# ADVANCED GROUP SEQUENTIAL DESIGNS
tutorial.tri <- update(tutorial.obf, P = 1, A = 1)
tutorial.asymm <- update(tutorial.obf, P = c(.5,1))
tutorial.hybrid <- update(tutorial.obf, test.type = "advanced",
                           epsilon = c(.5,1))
```

update.seqDesign Update a seqDesign Object

DESCRIPTION

Allows a new design to be created from an old model by providing only those arguments that need to be changed.

```
update.seqDesign(object, ..., evaluate=T, byValue=F)
update(object, ..., evaluate=T, byValue=F)
```

REQUIRED ARGUMENTS

object: a **seqDesign** object (or an object that inherits from "seqDesign").

OPTIONAL ARGUMENTS

...: any other arguments to **seqDesign** (or another function that created the object, e.g. **seqMonitor**). These must all be named, and may be abbreviated. Arguments in the previous fit; that is, in **object\$call**, can be removed by putting nothing on the right side of the **=**. For example, the argument **P=P** argument to be removed before a new object is created.

evaluate: if TRUE (the default), the new call is evaluated; otherwise, the call is returned as an unevaluated expression.

byValue: logical, if TRUE (the default) then updating takes place by value rather than symbolically. This has two effects: (1) If the original call included **seqDesign(sample.size=N, ...)** then the symbolic expression **N** is replaced by **object\$specification\$sample.size**, which contains the value of **N** when the object was originally created. Hence results will be consistent if **N** has changed. (2) Symbolic expressions in the call to **update** are replaced by their value, e.g. **update(object, P=foo)** creates a call to **seqDesign** with **foo** replaced by its value.

VALUE

either a new updated object, or else an unevaluated expression for creating such an object.

DETAILS

update is a generic function. Do **methods("update")** to see a list of currently-defined methods.

seqMonitor and **seqFixDesign** use **update.seqDesign** to create modified designs.

SEE ALSO

seqDesign, **seqMonitor** , **seqFixDesign** .

EXAMPLE

```
des <- seqDesign(alt.hypothesis = .1, nbr.analyses = 3)
update(des, test.type = "two.sided")
# For additional examples see help(seqDesign)
```

validate.seqtrial Validation Tests

DESCRIPTION

Performs one or more sets of validation tests for S+SeqTrial.

```
validate.seqtrial(file, test.loc, verbose = F,  
                  outfile, help = F)
```

OPTIONAL ARGUMENTS

file: vector of test file names. If **file** is not specified, all of the files located in the default **test.loc** directory will be executed.

test.loc: character string that specifies the location of the test files. The default directory is the **validate** subdirectory of the **seqtrial** module.

verbose: logical value that controls the printed output. If TRUE, each test expression will be printed, regardless of whether a test fails. Otherwise, only test expressions for failed tests will be printed. The default is FALSE.

outfile: character string that specifies the name of the output file to which test results will be written. By default, the output is written to the screen.

help: logical value that controls a display of valid test file names and descriptions. When **help=T**, a list of valid test files and their corresponding descriptions is displayed for the **test.loc** directory.

VALUE

logical value. This value is TRUE if all test pass, FALSE if any tests fail. test case expressions are written to either the screen or a specified output file. The test expressions to be written are controlled by the verbose argument.

DETAILS

This function is a front end to **validate**, designed specifically to handle validation for S+SeqTrial. Tests currently available are:

EmerFlem

Reproduce results published in Emerson and Fleming (1989).

PampTsia

Reproduce results published in Pampallona and Tsaiatis (1994).

WangTsia

Reproduce results published in Wang and Tsaiatis (1987).

fixed

Reproduce fixed-sample results from S-PLUS 2000 User's Guide.

Each test file contains a set of expressions that are evaluated and return a value of either TRUE or FALSE. The test files are organized by related functions. Use `validate` with `help=T` to display a list of test files for the default `test.loc` directory.

Each test is self-documenting in that it identifies the high-level function(s) being tested, the source of data, applicable references to published results, and a short description of the test case. All necessary data is also included as part of the test.

A test is considered to pass when each of the values being compared to published or otherwise known results are within a specified tolerance. The tolerance values vary among the tests—the tolerances used for a given test are specified immediately following the description. If one or more comparisons fail, the test fails.

Some of the tests take a considerable amount of time. Use of `verbose=T` provides an indication of progress. This lists the test being performed, including comments indicating how the test relates to the published references.

REFERENCES

Emerson, S. and Fleming, T. R. (1989). Symmetric group sequential test designs *Biometrics*, Vol 45, 905-923.

Pampallona, S. A. and Tsiatis, A. A. (1994). Group sequential designs for one-sided and two-sided hypothesis testing with provision for early stopping in favor of the null hypothesis. *Journal of Statistical Planning and Inference*, Vol 40.

Wang, S. K. and Tsiatis, A. A. (1987) Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics*, Vol 43, 193-199.

EXAMPLE

```
validate.seqtrial(help=T) # List of available tests
validate.seqtrial("EmerFlem", verbose=T) # Emerson and Fleming 1989
validate.seqtrial("PampTsia", verbose=T) # Pampallona & Tsiatis 1994
validate.seqtrial("WangTsia", verbose=T) # Wang & Tsiatis 1987
validate.seqtrial("fixed", verbose=T) # Fixed sample (one analysis time)
```


BIBLIOGRAPHY

- Armitage, P., McPherson, C. K., & Rowe, B. C. (1969). Repeated significance tests on accumulating data. *Journal of the Royal Statistical Society, Series A*, 132:235--244.
- Becker, R. A. & Cleveland, W. S. (1996). *S-Plus Trellis Graphics User's Manual*. MathSoft, Inc., Seattle, Murray Hill: Bell Labs.
- Cleveland, W. S. (1993). *Visualizing Data*. Hobart Press, Summit, NJ.
- Emerson, S. (1995). Stopping a clinical trial very early. *Biometrics*, 51:1152-1162.
- Emerson, S. (2000). S+SEQTRIAL: Technical Overview. Research Report No. 98, Data Analysis Products Division, MathSoft, Inc.
- Emerson, S. & Fleming, T. (1989). Symmetric group sequential test designs. *Biometrics*, 45:905--923.
- Emerson, S. & Fleming, T. (1990). Parameter estimation following group sequential hypothesis testing. *Biometrika*, 77:875--892, 1990.
- Emerson, S. & Kittelson, J. M. (1997). A computationally simpler algorithm for the UMVUE of a normal mean following a group sequential trial. *Biometrics*, 53:365-369.
- Jennison, C. & Turnbull, B. W. (1989). Interim analyses: the repeated confidence interval approach. *Journal of the Royal Statistical Society, Series B*, 51:305--361.
- Jennison, C. & Turnbull, B. W. (1999). *Group Sequential Methods with Applications to Clinical Trials*. CRC Press, Boca Raton, FL.
- Kittelson, J. M. and Emerson, S. (1999). A Unifying Family of Group Sequential Test Designs, *Biometrics*, 55:874-882.
- Lan, K. K. G. and DeMets, D. L. (1983). Discrete sequential boundaries for clinical trials, *Biometrika*, 70:659-663.
- Lan, K. K. G. & DeMets, D. L. (1989). Changing frequency of interim analysis in sequential monitoring. *Biometrics*, 45:1017--1020.

Bibliography

- O'Brien, P. C. & Fleming, T. R. (1979). A multiple testing procedure for clinical trials. *Biometrics*, 35:549--556.
- Pampallona, S. A. & Tsiatis, A. A. (1994). Group sequential designs for one-sided and two-sided hypothesis testing with provision for early stopping in favor of the null hypothesis. *Journal of Statistical Planning and Inference*, 40.
- Pampallona, S. A., Tsiatis, A. A., & Kim, K. M. (1995). Spending functions for the type I and type II error probabilities of group sequential tests. Technical report, Harvard School of Public Health.
- Pocock, S. J. (1977). Group sequential methods in the design and analysis of clinical trials. *Biometrika*, 64:191--199.
- Tsiatis, A. A., Rosner, G. L., & Mehta, C. R. (1984). Exact confidence intervals following a group sequential test. *Biometrics*, 40:797-803.
- Tsiatis, A. A., Rosner, G. L., & Trichler, D. (1985). Group sequential tests with censored survival data adjusting for covariates. *Biometrika*, 72:365--373.
- Whitehead, J. (1983). *The Design and Analysis of Sequential Clinical Trials*. Ellis Horwood, Chichester, UK.
- Whitehead, J. (1997). *The Design and Analysis of Sequential Clinical Trials*, 2nd Rev. Edition. John Wiley & Son Ltd, Chichester, UK.
- Whitehead, J. & Stratton, I. (1983). Group sequential clinical trials with triangular continuation regions. *Biometrics*, 39:227-236.
- Wang, S. K. & Tsiatis, A. A. (1987). Approximately optimal one-parameter boundaries for group sequential trials. *Biometrics*, 43:193--199.

INDEX

A

A argument to the seqDesign function 137
A boundary shape parameter 137
ad hoc stopping boundaries 316
Advanced tab 135
alpha 136
alpha argument to the seqDesign function 136
analysis time ordering 233
arbitrary stopping rules 159
ASN. See average sample number.
average sample number 176, 192
Average Sample Number plot type 177, 196

B

BAM 231, 236
Bayesian posterior probabilities 115, 344
Bayesian priors 115, 344
Bayesian scale 115, 167, 344
Bayesian scale parameters 141
beta 136
beta argument to the seqDesign function 136
bias adjusted mean 231, 236
Binomial Odds models 70
Binomial Proportion(s) models 69
boundary matrices 138, 159, 317
viewing and editing 138
boundary scale parameters 140

boundary scales 138, 164, 194
Bayesian 115, 344
error spending 115
fixed-sample P 114
input to functions 138
partial sum 112
sample mean 113
Z-statistic 114
boundary shape 133, 134
boundary shape parameters 137
boundary shift function 283

C

clinical trial structures 64
conditional futility scale 116, 117
conditional futility scale parameters 146
confidence intervals 163, 192, 196, 233, 236
constrained boundaries 130, 316
constraint scale 224

D

data-driven interim analyses 306
data-imposed stopping rules 306
Decision Boundaries plot type 172, 196, 200
design.family argument to the seqDesign function 138
design family 138
Designs to Plot groupbox 197

display.scale argument to the seqDesign function 194
display scale 168
Display Scale option 194, 196, 200

E

early stopping 130
epsilon 136, 253
epsilon argument to the seqDesign function 137
epsilon shift parameter 111, 253
error monitoring dialog 219
error spending scale 115
error spending scale parameters 152
estimators 231
ethical treatment of subjects 203
evaluating designs 161
exact.constraint argument to the seqDesign function 139
Exact Constraint boundary matrix 317
Exact Constraint matrix 138, 159

F

fixed argument 199
Fixed Sample P scale 167
fixed-sample P scale 114
functions, by category 345
futility scale parameters 146
futility scales 165

G

G argument to the seqDesign function 137
G boundary shape parameter 137
graphsheets function 199
group sequential designs 101

H

Hazard Ratio models 71
hybrid tests 110, 136, 239, 253

I

Include Reference option 197
Inference plot type 196, 237
inferences 229
interim analyses 132
number and timing 131

K

k Samples trial 66

L

linear regression 67
logistic regression 67
Log-Normal Median(s) models 69
lower hypothesis test 136, 253
lower power function 180, 191, 196

M

Max Constraint matrix 138, 159
maximum.constraint argument to the seqDesign function 139
maximum likelihood estimate 164, 231, 236
median unbiased estimate 231, 236
Min Constraint matrix 138, 159
minimum.constraint argument to the seqDesign function 139
minimum spacing 226
MLE 164, 231, 236
monitoring 272
reasons for 130, 203
statistical issues 204
monitoring a trial 201
MUE 231, 236
multiple arm trial 66

N

nbr.analyses argument to the seqDesign function 133
Normalized Z statistic scale 167
number of analyses 132

O

- one arm trial 64
- One Sample trial 64
- Open New Graphsheet option 199
- operating characteristics 188
- Operating Characteristics groupbox 136
- ordering outcomes 233
 - by analysis time 233
 - by sample mean 233
- Overlay Designs option 178, 183, 197, 199

P

- P argument to the seqDesign function 134, 137
- partial sum scale 112
- P boundary shape parameter 133, 134, 137
- plotting options 199
- point estimates 163, 192, 196, 231, 236
- Poisson Rate(s) models 70
- Poisson regression 67
- Power Curve plot type 182, 196
- power functions 180
- predictive futility scale 116, 117
- predictive futility scale parameters 146
- printed results 187
- probability models 68
 - Binomial Odds 70
 - Binomial Proportion(s) 69
 - Hazard Ratio 71
 - Log-Normal Median(s) 69
 - Normal Mean(s) 68
 - Poisson Rate(s) 70
- p-values 192, 235, 236

R

- Rao-Blackwell adjusted estimate 231, 236

R argument to the seqDesign

function 137

RBadj 231, 236

R boundary shape parameter 137

Regression trial 66

reporting results 229

response variability 204

results

printed 187

Results tab 187

S

sample mean ordering 233

sample mean scale 113, 164

sample size 188

sample size distribution 196

tabulating 176

sample size distributions 176

Saved Results groupbox 181, 185, 194

scale parameters 140

Bayesian scale 141

error spending scale 152

futility scales 146

Scale Parameters fields 140

sensitivity analysis 283, 307

seqDesign function 188

seqInference function 237

seqMonitor function 236, 272

seqPHSSampleSize function 72, 86

sequential design

ad hoc 316

Sequential Design groupbox 132

shift parameter 136, 253

Short Output option 188

significance level 136, 180

stochastic curtailment 116, 165

stopping boundaries 164

plotting 172, 173

printing 168

stopping probabilities 185, 192

Stopping Probability plot type 185, 197

stopping rules

- monitoring issues 130
- stoppping boundaries 130
- summary function 177, 181, 182, 185, 193, 237
- Summary Tables option 168, 181, 185, 191, 236
- superpose.design argument 198, 200
- support, technical 15

- T**
- technical support 15
 - outside USA and Canada 15
 - USA and Canada 15
- two arm trial 65
- Two Samples trial 65
- types of clinical trials 64
 - k Samples 66

- U**
- One Sample 64
- Regression 66
- Two Samples 65
- types of probability models 68

- Z**
- UMVUE 231
- unexpected toxicities 289
- uniform minimum variance
 - unbiased estimate 231
- unplanned analyses 261
- upper hypothesis test 136, 137, 253
- upper power function 180, 191, 196

- Z-statistic scale 114