An R Package for Group Sequential Boundaries Using Alpha Spending Functions

T. Charles Casper

Oscar A. Perez

February 20, 2006

1 Summary

Reboussin *et al.* [1] created a `FORTRAN` program, `ld98`, which performs computations related to group sequential analysis in clinical trials using spending functions. The original method was proposed by Lan and DeMets [2] and the method has also been described in [3]. These kinds of methods have been developed because of the ethical responsibility to monitor accumulating data during a clinical trial. The trial stops early if an interim analysis reveals a large enough difference between treatment groups. The Fortran program is interactive and has four options which compute: (1) bounds given analysis times and a spending function, (2) drift parameters corresponding to given power and bounds, (3) probabilities given times, bounds, and drift parameters, and (4) a confidence interval given times, bounds, and final test statistic value. The program consists of one main program, an input routine, 17 subroutines, and 5 functions related to probability and probability density calculations.

For this project we created an `R` package [5], `ldbounds`, which performs esentially the same tasks. The functions in the package are not interactive in the same way as

the Fortran program (prompting for inputs, etc.). This package has some advantages: it uses R's capability to work with the "whole object" as opposed to always using loops and also uses R's built in probability distribution calculations, it has better output graphics, and it allows for more freedom in specifying spending functions.

2 Introduction

In most clinical trials, data are accumulated over a period of time that could be years. In these types of experiments there is a lot at stake. The amount of money and other resources spent is very large. Also, humans are the subjects of the experiments, leading to a very sensitive ethical situation. For these reasons, a clinical trial should not go on longer than is necessary to show a significant result with the accumulating data. In fact, most clinical trials experience interim review of data by an independent Data and Safety Monitoring Board (DSMB). This board may decide to recommend early termination of the trial. However, significance relies on the expected frequency of certain events under the null hypothesis. If a trial is monitored at several analysis times, the probability of observing a "significant" result increases. It has been shown that [4], with 10 analysis times, the overall type I error rate of 5% increases to about 20%.

This is the motivation behind group sequential methods. The goal is to allow for several looks of the data throughout the course of a trial and still control the overall type I error rate. Prior to the alpha spending function method, other techniques had several limitations. Among these was the need to specify the number and exact times of all interim analyses before a trial began. The alpha spending function method, proposed by Lan and DeMets [2, 3], is much more flexible and interim analysis times can be specified or adjusted at any time during the trial. The basic concept is to define a function which determines how much of the total type I error can be "spent" at each analysis, based only on the previous analysis times and the amount of change in the specified function since the last time. It should be mentioned that these methods are

not always strictly followed by a DSMB, but instead they are used more as guidelines. Lan and DeMets also created a `FORTRAN` program to make computations related to alpha spending functions and group sequential boundaries. In more recent years, this program was revised and improved in terms of efficiency [1]. Our goal with this project was to create an `R` package to accomplish the same computations and even give more flexibility (see Section 1).

3 Methods

Formally, the group sequential method is the following: In a clinical trial, define a set of critical values $\{Z_C(k), k = 1, \ldots, K\}$, one critical value at each analysis, such that the trial is to continue if the observed statistic $Z(k)$ satisfies $|Z(k)| < Z_C(k)$ (two-sided test) and the overall type I error rate is α.

To see how the alpha spending function method works in the context of a one-sided test, let $B(t)$ be a standard Brownian motion process on $[0, 1]$. Next, let τ be the stopping time defined by $\tau = \inf\{t : B(t) > b\}$, for some fixed b. Now, let

$$\alpha^*(t) = P(\tau \leq t) = 2 - 2\Phi\left(b/\sqrt{t}\right) I\{0 < t \leq 1\}.$$

We notice that, if $b = z_{\alpha/2}$, the $1 - \alpha/2$ quantile of the standard normal distribution, then $\alpha^*(1) = \alpha$. Now, assume that $B(t)$ is only observed at times $\{t_i : i = 1, \ldots, K; 0 < t_1 < \cdots < t_K = 1\}$. Define b_1 to satisfy

$$P\left(B(t_1) > b_1\right) = P\left(\tau \in (0, t_1]\right) = \alpha^*(t_1).$$

Similarly, define $b_2, \ldots, b_K$ such that for $i = 2, \ldots, K$

$$P\left(B(t_i) > b_i,\ B(t_j) < b_j,\ j = 1, \ldots, i-1\right) = P\left(\tau \in (t_{i-1}, t_i]\right) = \alpha^*(t_i) - \alpha^*(t_{i-1}).$$

The idea is that the amount that α^* has increased since the last analysis time is alotted to the boundary crossing probability at the current analysis time. It is a very straightforward task to show that

$$b_1 = \sqrt{t_1}\Phi^{-1}\left(2\Phi\left(b/\sqrt{t_1}\right) - 1\right).$$

However, finding $b_2, \ldots, b_K$ involves computing densities iteratively [4], similar to

$$f_i(t) = \int_{-\infty}^{b_{i-1}} f_{i-1}(u) \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{1}{2}(t-u)^2\right\} \, du$$

This is done by numerical integration. Then, the result is integrated and the optimal boundary is found by searching for the limit of integration.

This situation assumes fixed b. The general method can account for $b(t)$. Of course, in practice, the function α^* (the alpha spending function) is specified. As long as this function is increasing and satisfies $\alpha^*(0) = 0$ and $\alpha^*(1) = \alpha$, the desired type I error level, then the function has the desired properties to lead to a boundary. We notice that at each step, the boundary can be calculated based on only the previous values and times and the change in α^*. Thus, the exact number and placement of analysis times need not be specified in advance, but instead can be specified at any time. Also, the method has only been described for a one-sided test. For a symmetric two-sided test, $\alpha^*/2$ can be used and then the negative of the boundary calculated is used for the lower boundary. In a similar way, the method can be applied with different α^*'s used for the upper and the lower boundary. There are many statistical techniques that are needed to apply these methods in cases when the data comes in different forms [1].

4 The `ldbounds` Package

The `R` package `ldbounds` is based on the `FORTRAN` program to do calculations related to group sequential boundaries using spending functions. It consists of 18 functions: 6 method functions, 10 utility functions and 2 main functions (See Figure 1). These main functions are called `bounds` and `drift`.

The package is useful in four different situations: (1) to calculate boundaries given spending functions, (2) to compute drift for a given power and boundaries, (3) to calculate probabilities for given boundaries, and (4) to compute the confidence interval at the end of the trial, adjusted for multiple looks. The function `bounds` is used in the first case and `drift` can help us in the other three cases.

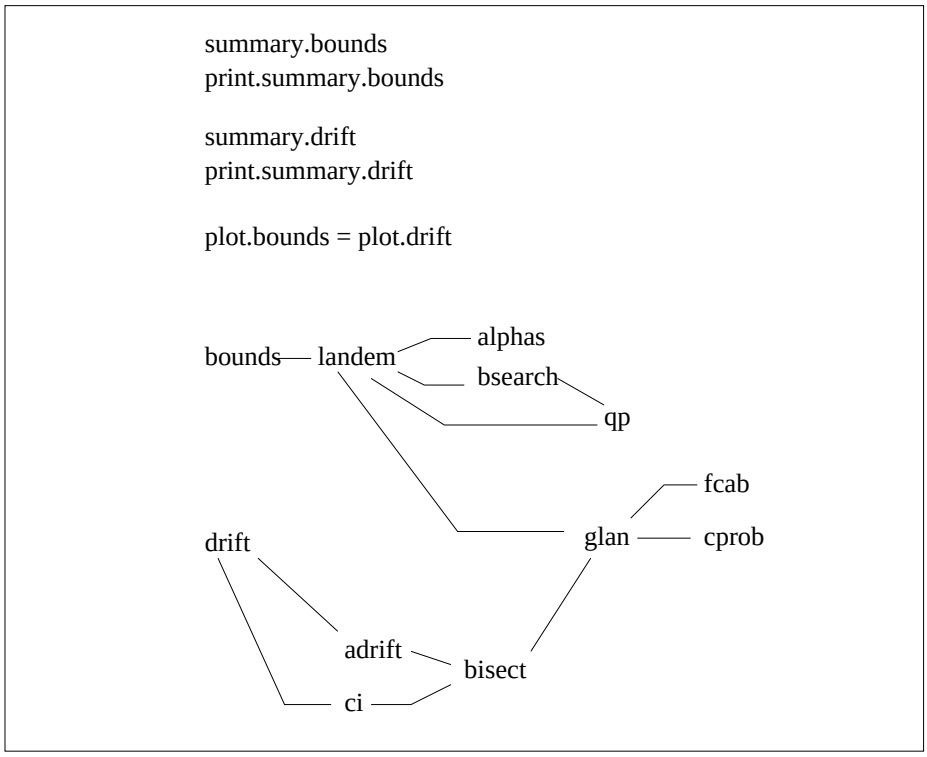

Figure 1: *Package* `ldbounds`. Schematic description of relationships between `ldbounds` functions.

4.1 The `bounds` Function

The function `bounds` determines group sequential boundaries for interim analyses of accumulating data in clinical trials using the Lan-DeMets alpha spending function method. These can be used as guidelines for early stopping of the trial. The following statement shows how to specify a call to bounds:

```
bounds(t, t2 = t, iuse = 1, asf = NULL,
       alpha = 0.05, phi = rep(1, length(alpha)),
       ztrun = rep(8, length(alpha)))
```

Here `t` corresponds to the vector of analysis times, which must be increasing and in (0,1]; `t2` is the second time scale, usually in terms of amount of accumulating information, by default `t2` is the same as `t`; `iuse` is a vector of the type of alpha

spending function(s) to use for lower and upper bounds, respectively (in the two-sided case), the program allows the user to specify the spending function from four different predefined options or the user can also specify a new spending function in which case `asf` must be used; `alpha` is a vector of type I errors, in two-sided situations, these correspond to the amount allocated to the lower and upper boundaries, respectively. The overall alpha must be greater than 0 and less than or equal to 1. `phi` is a vector of values used when `iuse`=3 or 4, and `ztrun` is a vector of values specifying where to truncate lower and upper boundaries, respectively, with a default of $(-8, 8)$ (or just 8 for one-sided), which is essentially no truncation.

The function `bounds` generates an object of class "bounds" which contains data about the boundaries calculated and exit probabilities, among other things. A nice way to extract information, as with most `R` object classes, is by using the `summary` and `print` methods for the "bounds" class.is can be done by setting `digit` to the desired number of digits.

To illustrate the use of `bounds` we present some examples from Reboussin, *et. al.* [1]. Using 5 equally spaced interim analyses with two-sided O'Brien-Fleming boundaries and $\alpha = 0.05$, we have:

```
> time <- seq(0.2, 1, length = 5)
> obf.bd <- bounds(time, iuse = c(1, 1), alpha = c(0.025, 0.025))
> summary(obf.bd)

Lan-DeMets bounds for a given spending function


n =  5
Overall alpha:  0.05


Type: Two-Sided Symmetric Bounds
Lower alpha:  0.025
Upper alpha:  0.025
```

```
Spending function:  O'Brien-Fleming

Boundaries:
   Time    Lower   Upper    Exit pr.   Diff. pr.
1   0.2  -4.8769  4.8769  1.0777e-06  1.0777e-06
2   0.4  -3.3569  3.3569  7.8830e-04  7.8723e-04
3   0.6  -2.6803  2.6803  7.6161e-03  6.8278e-03
4   0.8  -2.2898  2.2898  2.4424e-02  1.6807e-02
5   1.0  -2.0310  2.0310  5.0000e-02  2.5576e-02
```

The `plot` method for class "bounds" can be used to check the plot of the boundaries calculated. (See Figure 2).

```
> plot(obf.bd)
```

If we want to calculate the bounds for the one-sided case, we have:

```
> time <- seq(0.2, 1, length = 5)
> obf.bd <- bounds(time)
> summary(obf.bd)

Lan-DeMets bounds for a given spending function

n =  5
Overall alpha:  0.05

Type: One-Sided Bounds
alpha:  0.05
Spending function: O'Brien-Fleming

Boundaries:
   Time   Upper    Exit pr.   Diff. pr.
```

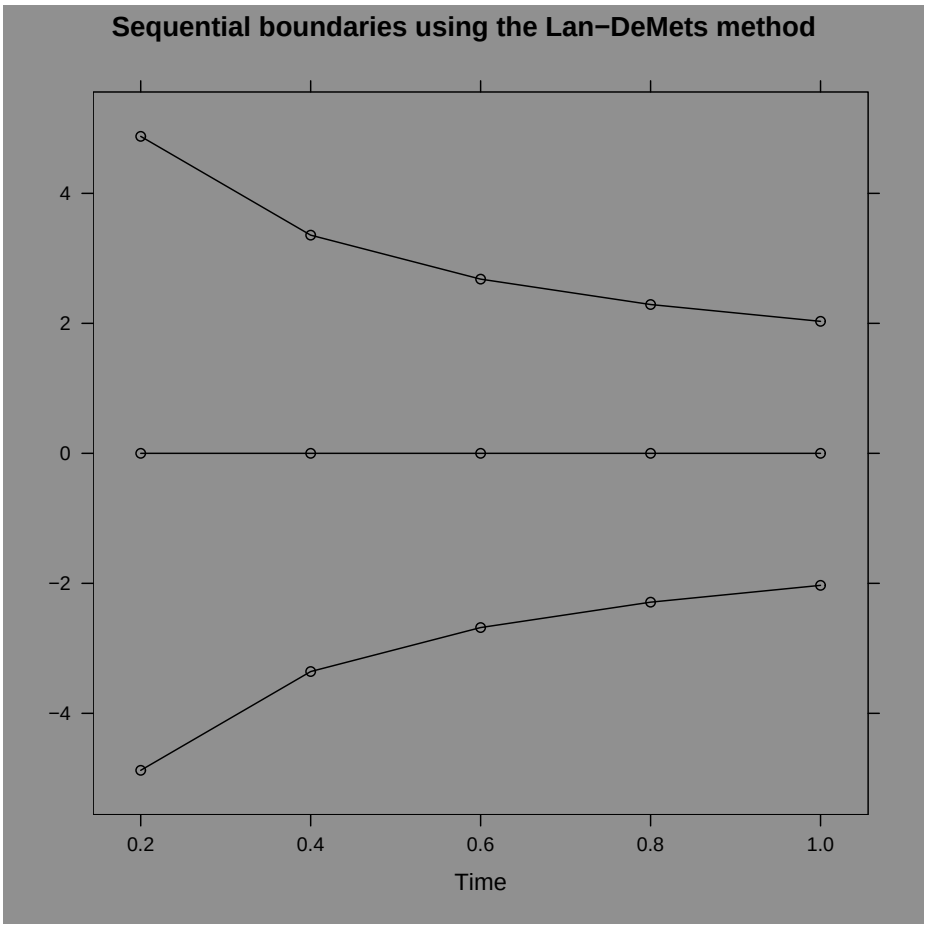

Figure 2: *Boundaries.* Two-sided plot for `plot(obf.bd)`.

```
1    0.2  4.2292  1.1726e-05  1.1726e-05
2    0.4  2.8881  1.9419e-03  1.9302e-03
3    0.6  2.2981  1.1396e-02  9.4545e-03
4    0.8  1.9618  2.8430e-02  1.7033e-02
5    1.0  1.7397  5.0000e-02  2.1570e-02
```

Here the boundaries are shorter than in the two-sided case, which is of course a consequence of specifying the same `alpha` in both cases. We can also get the boundaries plot again (See Figure 3).

```
> plot(obf.bd)
```

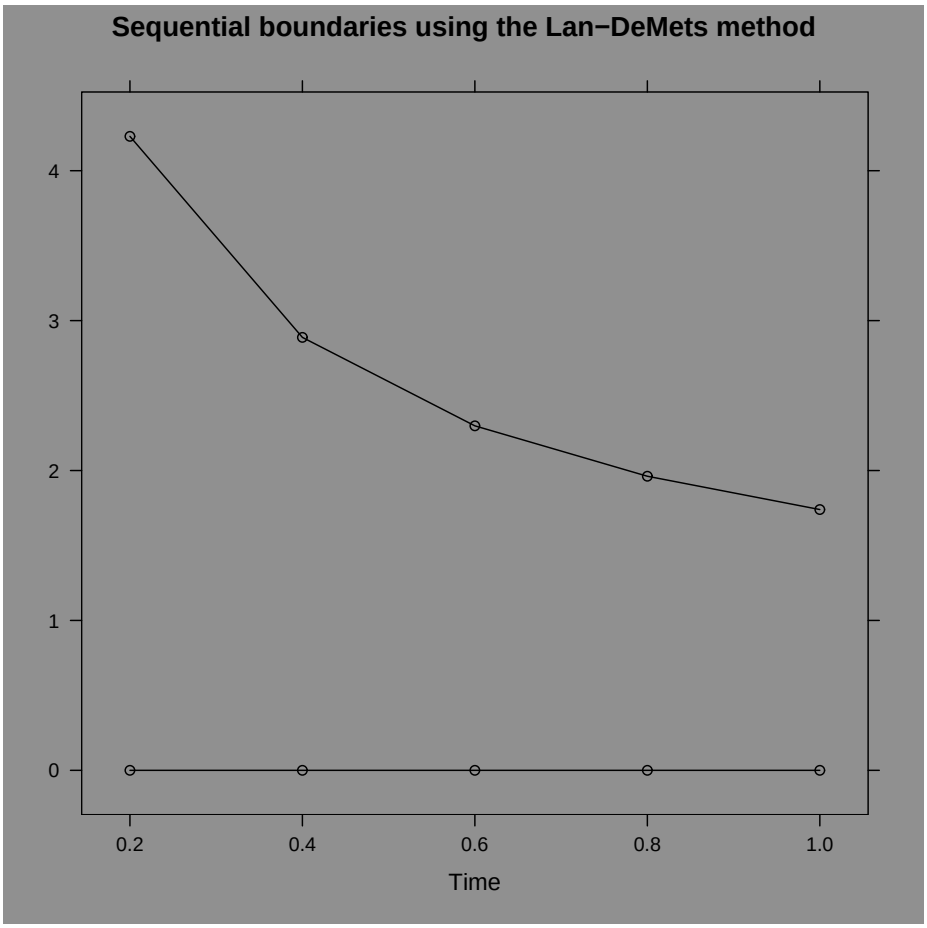

Figure 3: *Boundaries II.* One-sided plot for `plot(obf.bd)`.

4.2 The `drift` Function

The `drift` function was designed to calculate different things, depending on the arguments specified. It calculates drift (effect) if the power and the bounds are given; it also can calculate the confidence interval at the end of the trial if the last value of the standardized test statistic is provided. Finally, it is able to return the power and other probabilities given drift for specified boundaries. The function has the following parameters:

```
drift(za = -zb, zb, t, t2 = t, pow = NULL,
      drft = NULL, conf = NULL,
```

```
        zval = zb[length(zb)])
```

Here `za` is the vector of lower boundaries and symmetric to `zb` by default; `zb` is the vector of upper boundaries; `t` is the vector of analysis times, which as in `bounds` must be increasing and in (0,1]; `t2` is the second time scale, `pow` is the desired power (in case of wanting this parameter and drift is not specified); `drft` is the true drift (i.e. treatment effect when t=1); `conf` is the confidence level when a confidence interval for drift is wanted, and `zval` is the final observed Z statistic (i.e. when trial is stopped), it is required when a confidence interval is requested.

Similarly to `bounds`, the function `drift` generates an object of class "drift" which contains data about the boundaries, exit probabilities, and the corresponding calculations (depending on what was requested). As in `bounds`, there is a nice way to extract the information that it contains by using the `summmary` and `print` methods.

We now show some applications of `drift`. First we want to calculate the probabilities associated with a set of bondaries given drift.

```
> time <- c(0.13, 0.4, 0.69, 0.9, 0.98, 1)
> upper <- c(5.3666, 3.7102, 2.9728, 2.5365, 2.2154, 1.9668)
> drift.pr <- drift(zb = upper, t = time, drft = 3.242)
> summary(drift.pr)

Lan-DeMets method for group sequential boundaries


n =  6


Boundaries:
   Time    Lower   Upper
1  0.13  -5.3666  5.3666
2  0.40  -3.7102  3.7102
3  0.69  -2.9728  2.9728
4  0.90  -2.5365  2.5365
```

```
5  0.98  -2.2154  2.2154
6  1.00  -1.9668  1.9668

Drift parameters:  3.242
   Time       Lower       Upper    Exit pr.  Cum exit pr.
1  0.13  3.1695e-11  1.3483e-05  1.3483e-05    1.3483e-05
2  0.40  4.1900e-09  4.8468e-02  4.8468e-02    4.8481e-02
3  0.69  7.1472e-09  3.4279e-01  3.4279e-01    3.9127e-01
4  0.90  8.8185e-09  3.1827e-01  3.1827e-01    7.0954e-01
5  0.98  2.2251e-08  1.3325e-01  1.3325e-01    8.4279e-01
6  1.00  6.4894e-08  5.6851e-02  5.6851e-02    8.9964e-01
```

Another common task is to calculate a confidence interval at the end of the trial. In order to do this, we need, in addition to the boundaries, the last value of the standardized test statistic `zval`, equal to 2.82 in this example.

```
> time <- c(0.2292, 0.3333, 0.4375, 0.5833, 0.7083, 0.8333)
> upper <- c(2.53, 2.61, 2.57, 2.47, 2.43, 2.38)
> drift.ci <- drift(zb = upper, t = time, conf = 0.95, zval = 2.82)
> summary(drift.ci)

Lan-DeMets method for group sequential boundaries

n =  6

Boundaries:
     Time  Lower  Upper
1  0.2292  -2.53   2.53
2  0.3333  -2.61   2.61
3  0.4375  -2.57   2.57
4  0.5833  -2.47   2.47
```

```
5  0.7083  -2.43   2.43
6  0.8333  -2.38   2.38

Confidence interval at the end of the trial:

Confidence level:  0.95
Last Z value:  2.82
 95 % confidence interval: ( 0.1716815 , 4.504665 )
```

The following example shows the computing of boundaries using `bounds`, so they can be used in `drift`. In this example the power is given and we want to get the drift.

```
> time <- seq(0.2, 1, length = 5)
> obf.bd <- bounds(time, iuse = c(1, 1), alpha = c(0.025, 0.025))
> drift.dr <- drift(obf.bd$lower.bounds, obf.bd$upper.bounds, time,
+     pow = 0.9)
> summary(drift.dr)

Lan-DeMets method for group sequential boundaries

n =  5

Boundaries:
   Time    Lower   Upper
1   0.2  -4.8769  4.8769
2   0.4  -3.3569  3.3569
3   0.6  -2.6803  2.6803
4   0.8  -2.2898  2.2898
5   1.0  -2.0310  2.0310
```

```
Power :  0.9


Drift:  3.278782
```

	Time	Lower	Upper	Exit pr.	Cum exit pr.
1	0.2	1.1252e-10	0.00032414	0.00032414	0.00032414
2	0.4	2.8064e-08	0.09938662	0.09938665	0.09971079
3	0.6	8.5565e-08	0.34658009	0.34658018	0.44629097
4	0.8	7.4817e-08	0.29965581	0.29965589	0.74594686
5	1.0	4.0215e-08	0.15405261	0.15405265	0.89999951

5 Discussion

The `R` package `ldbounds` is able to perform the same computations as the `FORTRAN` program. It also allows for more flexibility for specifying alpha spending functions. Many improvements need to be made in the package in the near future. The function `drift` needs to include a better way to take its arguments from the values supplied by `bounds`. This will make it easier for the user. Other possible improvements might include better ways to specify arguments, more efficient argument defaults, and better graphics.

References

[1] Reboussin DM, DeMets DL, Kim K, Lan KKG. Computations for group sequential boundaries using the Lan-DeMets spending function method. *Controlled Clinical Trials*. 2000; 21:190-207.

[2] Lan KKG, DeMets DL. Discrete sequential boundaries for clinical trials. *Biometrika*. 1983; 70:659-63.

[3] DeMets DL, Lan G. "The alpha spending function approach to interim data analyses" in *Recent Advances in Clinical Trial Design and Analysis*, ed. PF Thall. Kluwer Academic Publishers, Boston; 1995.

[4] Armitage P, McPherson CK, Rowe BC. Repeated significance tests on accumulating data. *Journal of the Royal Statistical Society*. 1969; 132:235-44

[5] R Development Core Team (2005). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.