A SAS Macro for Calculating Confidence Limits and P-values Under Simon's Two-Stage Design

Alex Karanevich and Michael Ames, EMB Statistical Solutions

ABSTRACT

Simon's two-stage designs are popular single-arm, binary-endpoint clinical trials that include a single interim analysis for futility. One is typically interested in the proportion of successes (response rate), but SAS procedures (e.g. FREQ) do not provide p-values or confidence limits for this parameter that include the multiplicity adjustment for when such a trial proceeds beyond the first stage. This quick tip provides derivations (along with a SAS macro) for calculating such a trial's p-value and associated confidence limits. The macro requires the user to input the design parameters for the trial, as well as the total observed successes.

INTRODUCTION

Simon's two-stage design is a clinical trial design with a single arm, a single binary endpoint (usually success/failure), and a single interim look for futility. The design is typically used to test the null hypothesis that the probability of observing a success in a single patient, p, is less than or equal to some value p_0, or $H0: p \leq p_0$. To do this, first recruit and treat n_1 patients and note the number of patients who are judged "successes" (e.g. cancer remission, loss of sufficient body fat). If the number of successes is less than or equal to a pre-specified value r_1, stop the trial early for futility. Otherwise, recruit an additional n_2 patients (for a total of $N = n_1 + n_2$ patients) and note the total number of successes out of the N total patients. If this total number of successes is less than or equal to a pre-specified value r_2, then fail to reject $H0: p \leq p_0$. Otherwise, if the total number of successes exceeds r_2, then reject $H0$. The design is shown graphically in Figure 1.
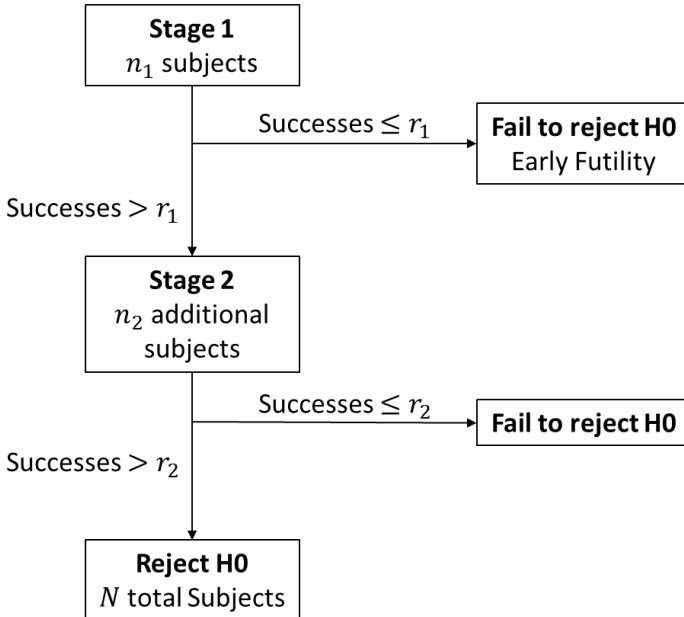

Figure 1. Diagram of Simon's two-stage design

For a given p_0 and type I error rate α, constructing a Simon's two-stage design is straightforward (i.e. using http://cancer.unc.edu/biostatistics/program/ivanova/SimonsTwoStageDesign.aspx which will specify r_1, r_2, n_1, and N for some inputted statistical power and alternative hypothesis), but any post-trial analyses on the parameter p, such as confidence intervals or p-values, are somewhat complicated by the interim futility analysis in the design.

Fortunately, a closed-form solution exists to this problem. Koyama and Chen (2008) produce a method of calculating the exact p-value for a Simon's two-stage design, and note that this p-value will always be at least as small as the p-value obtained by naively ignoring the interim step of the design (i.e. simply performing an exact binomial test on the number of successes out of N or n_1). To the best of our knowledge, no published SAS macro or procedure produces this p-value and corresponding confidence intervals – creating such a macro is the purpose of this paper.

MATHEMATICS AND NOTATION

Let the number of patients with a successful outcome in stage i of the trial be denoted as X_i. Then the total number of successes is $X = X_1 + X_2$, where $X_1 \sim Binomial(n_1, p)$ and $X_2 \sim Binomial(n_2, p)$ if $X_1 > r_1$, and $X_2 = 0$ otherwise. When testing the null hypothesis $H0: p \leq p_0$, if the trial terminates early in the first stage, then $X = X_1$ and exact p-values and confidence intervals can be calculated using exact binomial methods. However, if the trial does not terminate early, then one must incorporate the knowledge that $X_1 > r_1$ when calculating the p-value.

We first find the probability $f(x) = \Pr(X = x | X_1 > r_1)$. The p-value for $H0$ when observing $K > r_1$ total successes in the trial is then $\sum_{x=K}^{N} f(x)$. Since X_1 and X_2 are independent, we see that:

$$\Pr(X = x | X_1 = x_1 > r_1) = \Pr(X_2 = x - x_1) \Pr(X_1 = x_1)$$

$$= \binom{n_1}{x_1} \binom{n_2}{x - x_1} p_0^x (1 - p_0)^{N-x}$$

If we relax the condition $X = x > r_1$ to be solely $X_1 > r_1$, we can sum over all possible choices of x_1 to obtain:

$$\Pr(X = x | X_1 > r_1) = \sum_{x_1 = \max(r_1 + 1, x - n_2)}^{\min(x, n_1)} \binom{n_1}{x_1} \binom{n_2}{x - x_1} p_0^x (1 - p_0)^{N-x}$$

Where the upper and lower bounds of the sum are all the various ways x can be split between x_1 and x_2. Finally, the p-value when observing $K > r_1$ successes at the end of the trial can be obtained by

$$\Pr(X \geq K | X_1 > r_1) = \sum_{x=K}^{N} \Pr(X = x | X_1 > r_1)$$

$$= \sum_{x=K}^{N} \left\{ \sum_{x_1 = \max(r_1 + 1, x - n_2)}^{\min(x, n_1)} \binom{n_1}{x_1} \binom{n_2}{x - x_1} p_0^x (1 - p_0)^{N-x} \right\},$$

which is straightforward to implement as a SAS macro.

MACRO FOR FINDING P-VALUES

To calculate the stage 2 p-value, the user first needs to input the parameters for the design (N, n_1, r_1, p_0) and the observed total number of successes. From there, one can use a nested pair of do-loops to calculate the desired p-value. An example of code that produces this p-value is given below. This macro works within a larger macro provided in the next section. It can be modified by uncommenting the second and second-to-last lines to generate a dataset "pval" that contains the variable "p_val", which is the calculated p-value. Note the input "num_successes" is the observed total number of successes and must be greater than the input "r1"; otherwise, the trial did not reach stage 2 and this p-value is not appropriate:

```
%macro Simon_p(num_success=, n=, n1=, r1=, p0=);
*data pval; *uncomment to produce a data set along with run statement;
```

```
    p_val = 0;
    do X = &num_success to &n;
              do X1=max((&r1+1),(X-(&n-&n1))) to min(X, &n1);
                   X2 = X - X1;
                   pr_x = PDF('BINOMIAL', X1, &p0, &n1) *
                        PDF('BINOMIAL', X2, &p0, &n-&n1);
                   p_val = p_val+pr_x;
                   retain p_val;
              end;
         end;
  *run; *uncomment to produce a data set.;
  %mend;
```

MACRO FOR FINDING CONFIDENCE LIMITS

Now that we can calculate the p-value, we can use it to calculate confidence limits at some type I error rate α. Similar to the Clopper-Pearson method, the method proposed by Koyama and Chen (2008) is to adjust the assumed value of p_0 until a p-value of α is obtained. This value of p_0 corresponds to the lower confidence limit of the test (recall the test is one-sided). Similarly, adjusting the assumed value of p_0 until a p-value of $1 - \alpha$ is obtained provides an upper confidence limit.

The confidence interval established this way is really two α-level confidence limits, meaning the overall confidence interval is a $1 - 2\alpha\%$ confidence interval. Because Simon's two-stage designs are one-sided in nature, a two-sided confidence interval calculated at the $1 - \alpha\%$ ($\alpha/2$ confidence limits) may not correspond to the exact p-value and could result in rejection of $H0$ according to the p-value but not the confidence limit.

A complete macro that calculates and reports the p-value and two one-sided confidence limits is provided below. In addition to the specifications needed for the p-value (as above), the user also needs to specify the α-level for the confidence limits (assuming both upper and lower are at the same α-level). The macro creates a report giving the maximum likelihood estimate (MLE) of p, the p-value, and the confidence limits. An error is reported if the user attempts to provide a scenario that terminates at the end of stage 1 for futility:

```
%macro Simon_CI(num_success=, alpha=, n=, n1=, r1=, p0= );
    data confint;
        length errors $100; *To catch generic errors;
        phat_estimate = &num_success/&n; *MLE for p;
        if &num_success > &r1 then do; *Only proceed if successes > r1;
           errors = "None!";

           **Calculate the observed p-value;
           %Simon_p(num_success=&num_success, n=&n, n1=&n1, r1=&r1, p0=&p0);
           p_value = p_val;

           **Calculate the lower confidence limit at alpha specified;
           p_val = 0;
           conf_L = 0;
           do while (p_val < &alpha); *start at 0. Increase until p>alpha;
                *Below sets the tolerance for the accuracy of the limit,
                defaulted to 0.0001. For more accuracy, choose a value
                closer to 0;
                conf_L = conf_L + 0.0001;
                %Simon_p(num_success=&num_success, n=&n, n1=&n1, r1=&r1,
                     p0=conf_L);
           end;
```

```
***Calculate the upper confidence limit at alpha specified;
    p_val = 1;
    conf_U = 1;
    do while (p_val > (1-&alpha));*decrease from 1 until p<(1-alpha);
        conf_U = conf_U - 0.0001;*accuracy threshold: -0.0001;
        %Simon_p(num_success=&num_success, n=&n, n1=&n1, r1=&r1,
            p0=conf_U);
    end;
end;
***Generate report;
    if &num_success <= &r1 then do;
        p_value = .;
        conf_L = .;
        conf_U = .;
        errors = "Failed at stage 1, use exact binomial methods";
    end;
    output;
    keep phat_estimate p_value conf_L conf_U errors;
run;

%let conf_level=%SYSEVALF((1-&alpha)*100);
%let conf_level2=%SYSEVALF((1-2*&alpha)*100);

proc print data=confint;
    title "Simon's two-stage p-value and one-sided &conf_level%
    confidence limits";
    title2 "N total: &n, Total successes: &num_success";
run;
%mend;
```

EXAMPLE

Koyama and Chen (2008) provide an example in their paper which we will mimic using the above macros. Consider an example design with $n = 29$, $n_1 = 10$, and $r1 = 1$, that is testing $H0: p \leq 0.1$ at $\alpha = 0.05$, and observes $X = 6$ total successes by the end of the trial. We call the macro:

```
%Simon_CI(num_success=6, alpha=0.05, n=29, n1=10, r1=1, p0=.1);
```

which results in the report shown in Figure 2, matching the results of the example in Koyama and Chen.

Simon's two-stage p-value and one-sided 95% confidence limits
N total: 29, Total successes: 6

Obs	errors	phat_estimate	p_value	conf_L	conf_U
1	None!	0.20690	0.047086	0.1016	0.4007

Figure 2. The report generated by the macro for the given example from Koyama and Chen

CONCLUSION

In this paper we provided a macro for calculating p-values and confidence intervals under Simon's two-stage design when the design does not stop early for futility. If the trial does stop early for futility, then exact binomial methods may be used instead. The macro requires the user to enter the design parameters, the number of observed successes, and the desired type-I error rate. Finally, we verified the macro provides equivalent results for an example in the literature.

REFERENCES

Koyama, Tatsuki. Chen, Heidi. 2008 July 20. Proper inference from Simon's two-stage designs. *Stat Med.* 27(16): 3145-54.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Alex Karanevich
EMB Statistical Solutions
akaranevich@embstats.com