

The logo for Biogen Idec, featuring the company name in a white, lowercase, sans-serif font. The text is contained within a grey rectangular box with a white border and a small notch on the top-left corner. A solid yellow square is positioned to the right of the logo box.

biogen idec

Writing SAS[®] Functions without SAS/TOOLKIT[®]

Jerry W. Lewis, PhD

Jerry.Lewis@biogenidec.com

BASUG 28 Mar 2008

Overview

Extremely simple but useful idea:

If the SAS code produced by a macro contains no semicolons, then the macro can behave like a SAS function, in that it can be called within a larger SAS expression.

Illustrated with a macro “function” to calculate Clopper-Pearson “exact” binomial confidence limits.

Background

- A confidence interval is a random interval (L, U) that is guaranteed to contain an unknown parameter θ with at least the pre-specified probability, regardless of the true value of θ . E.g., a 95% CI for p has
$$\Pr(L \leq p \leq U) \geq 0.95$$
- Normal theory binomial intervals only satisfy this condition asymptotically
- Clopper and Pearson (1934) obtained an “exact” confidence interval

Clopper-Pearson Confidence Interval

- Observe s successes in n trials
- 1-sided
 - LCL = **BETAINV**($1-conf, s, n-s$)
 - UCL = **BETAINV**($conf, s+1, n-s$)
- 2-sided
 - LCL = **BETAINV**($(1-conf)/2, s, n-s$)
 - UCL = **BETAINV**($1-(1-conf)/2, s+1, n-s$)

Calling Syntax

`%BinomCL (s, n, direction=1,conf=0.95,sides=1)`

where s = number of successes
 n = number of trials
 conf (optional) = confidence level
 sides (optional) = 1 or 2

and

1-sided 95% CL means 5% in each tail
2-sided 95% CL means 2.5% in each tail

Macro returns a numeric value or missing value

Usage Examples

- `UCL = %BinomCL(3,20);`
- `LCL = %BinomCL(s,n,direction=-1);`
- `UCL = %BinomCL(s,s+f);`
- `width = %BinomCL(s,n,conf=1-alpha,sides=2) -
%BinomCL(s,n,conf=1-alpha,sides=2,direction=-1);`
- `IF %BinomCL(s,n) > 0.3 THEN`

Advantages

- Easy to read calling program
- No dedicated data step
- No reserved SAS variable names
- Can handle many cases at one time
- Handle additional calculations immediately
- No need to remember formulas
- Closed reusable modules
 - Create, document and validate code once
 - Single code copy (%INCLUDE or autocall)

Implementation Issues

- parameter checking
- algorithm branches
- timing of variable evaluation
- support SAS expressions in arguments

Macros Process Before the Data Step

- Values of SAS variables are not available to the macro, only to the Data Step
- Arguments driving macro logic must be passed as constant expressions (e.g. *sides* and *direction*)
- Macro use of arguments not passed as constants should be wrapped in parentheses to allow for the possibility of unevaluated expressions (e.g. *s* and *conf*)

Parameter Checking

- Arguments passed as constants can be checked by the macro
 - %SYSEVALF allows constant expressions
- Arguments passed as SAS variables can only be checked by SAS functions (since we are producing only part of one SAS statement)
- IFN is an IF function that allows SAS branching
 - IFN evaluates both branches regardless of value, and so may generate useless warnings in the Log
e.g. BETAINV requires $a, b > 0$ (excludes $s=0$ and $s=n$)

Design Decisions

- Pass *s*, *n* and *conf* as SAS expressions for flexibility
- Pass *direction* and *sides* as constants
 - Contextually determined, so flexibility rarely needed
 - Simplified branching without nested IFN calls
- Pass *s* and *n* as positional arguments since always needed
- Pass *conf* and *sides* as optional arguments with reasonable defaults
- *direction* could go either way; optional avoids the need to enter the default value, but requires the name to be given in order to override

Resulting SAS Code

```
%MACRO BinomCL(s,n,direction=1,conf=0.95,sides=1);  
  %IF &sides=1 OR &sides=2 %THEN %DO;  
    %IF &sides=2 %THEN %LET conf = 1-(1-(&conf))/2;  
    %IF &direction<0 %THEN /* lower confidence limit */  
      IFN(&s=0,0,BETAINV(1-(&conf),&s,&n-(&s)+1));  
    %ELSE /* upper confidence limit */  
      IFN(&s=&n,1,BETAINV(&conf,&s+1,&n-(&s)));  
  %END;  
  %ELSE .  
%MEND BinomCL;
```

Formulas: Other Confidence Limits

- Poisson rate
- Normal standard deviation
- Normal CV (*aka* RSD)
- Normal C_p capability index
- Normal C_{pk} capability index (1-sided spec)

Formulas: Other Applications

- Distribution of r^{th} of n ordered observations from a specified distribution
- Skellam distribution (difference of 2 Poissons)
- Normal 1-sided tolerance factors
- Distribution Free tolerance limits
- *Other? (Discussion if time permits)*

Questions/Discussion?
