PharmaSUG China 4th Conference, 2015

Clinical trials optimization: Monte Carlo Simulation modeling and SAS applications

Ye Meng, PPD Inc., Beijing

ABSTRACT

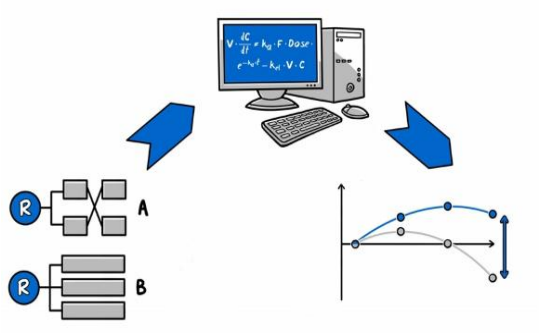
Modeling and clinical trial simulation is a tool that is being used by pharmaceutical companies and FDA to improve the efficiency of drug development. Monte Carlo Simulation is a modern and computationally efficient algorithm. Therefore it is a brilliant technique in terms of patient recruitment process and dose calculation in clinical design. The purpose of this paper is to describe how Monte Carlo simulations are tasked with evaluating parameter distributions and its applications in PROC MI. Here we also describe two approaches, BY command / Proc IML each with a presented example in SAS that leads to short and fast simulation programs that can be used in many practically situations.

The SAS V9 products used in this paper are SAS BASE®, SAS/STAT®, and SAS/GRAPH® on a PC Windows® platform.

INTRODUCTION

Drug development is not without risk, in fact many drug candidates are failing. Modeling and simulation is a risk management tool that could help to predict potential efficacy and safety outcome of clinical trials. The simulation explores how different trial designs will perform to detect expected drug effect. Father more, it shows the impact of differences of dose regiment, patient profile, sample size, drug duration under a multitude of uncertainties*[1]*. In the context of available scientific data, this can assess and evaluate potential risks and outcomes of trials before making huge investments. Modeling and simulation can be applied throughout the many phases of drug development. Therefore it can help provide guidance in decision making by quantifying and evaluating potential choices in clinical trials.

Monte Carlo methods, are sampling techniques that draw pseudo-random samples from specified probability distribution. In other words, Monte Carlo methods are numerical methods that utilize sequence numbers of random numbers to perform statistical simulations.*[2]*



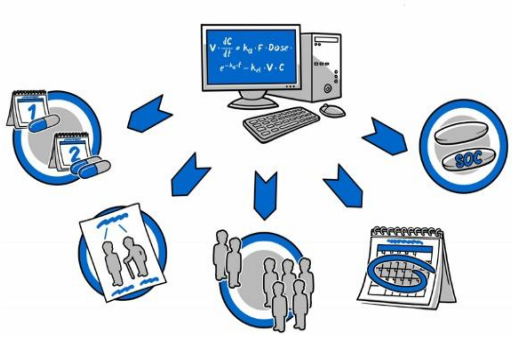


Figure 1. Clinical Trial Simulation (CTS) explores how different trial designs performance to detect the expected drug effect. Optimization including dose regiment, patient profile, sample size, trial duration and current standard of care (SOC) of patients in clinical trial simulation.

SIMULATING UNIVARIATE DATA IN DATA STEP:

In Principle, the simulation progress includes the following steps:
Step 1: Create a parametric model, y = f(X1, X2, …, Xq)
Step 2: Generate a set of random inputs, Xj1, Xj2, …, Xjq
Step 3: Evaluate the model and store the results as Yj
Step 4: Repeat steps 2 and 3 for i = 1 to n
Step 5: Analyze the results using histograms, summary statistics, confidence intervals.

There are different ways of organizing simulations in SAS. Take simulating univariate data for a quick example. A simple simulation might investigate the distribution of the sample mean of a sample of size 10 that is drawn randomly from the normal distribution on [5, 3]. First approach shows how DATA step estimates the mean 1000 times for sample size of 10 per iteration, as shown in the following example.

```
/* prevents log window from overflowing */
options nonotes;
/* set the parameters */
%let N = 10;
%let NumSamples = 1000;
%let mu = 5;
%let sigma = 3;
%let seed = 12345;

/******BY STATMENT APPROACH *****/
data Simulation(keep=SampleID x);
do SampleID = 1 to &NumSamples;          /* 1. create many samples */
 do i = 1 to &N;                 '       /* sample of size &N */
    x = rand("Normal", &mu, &sigma);     /* X ~ N(5, 3) */
    output;
 end;
end;
run;

proc means data=Simulation noprint;
 by SampleID;                            /* 2. compute many statistics */
 var x;
 output out=OutStats mean=SampleMean stderr=se tm;
run;

proc univariate data=OutStats;              /* 3. analyze the sampling
distribution of the statistics */
 histogram SampleMean;
```

```
run;

data summ;                                  /* 4. summarizing calculation */
file print;
set OutStats end=eof;
retain coverage 0;
 coverage=coverage+ SampleMean/&NumSamples;
     if eof then do;
     put  'coverage=' coverage 8.5;
     time=time();
     put 'END BY PROCESSING:' time=time 16.6;
     end;
run;
```

Note: The first step simulates one random sample with a Do Loop around. The second step is to compute the statistics for each sample. Then use BY statement in the procedure and performing statistics calculation.

The BY processing restructure the above simulation algorithm a little as the schema below:

1. Simulate many random samples of size N from a statistical model.
2. Compute a statistic for each sample.
3. Examine the union of the statistics, which approximates the sampling distribution of the statistic and tells you how the statistic varies due to sampling variation.

```
coverage= 5.05619
END BY PROCESSING:time=80512.187      80512.187000
```

Result 1. Coverage mean and time on distribution X ~ N(5, 3) after repeating 1000 times

The Mean finally converges to 5.05619 after 1000 times sampling. The programs were run in SAS 9.2 on computer with 200 GHz processor and 2.00 GB of RAM, operating under Windows XP.

SIMULATING UNIVARIATE DATA IN SAS/IML SOFTWARE:

The SAS/IML language made the simulation steps simple and easy. The following sample code is an example. *[3]*

```
/******SIMULATING UNIVARIATE DATA IN SAS/IML *****/
proc iml;
call randseed(&seed);
x = j(1, &NumSamples);              /* allocate vector or matrix */
call randgen(x, "Normal", &mu, &sigma);
```

The sample code calls two functions – RANDSEED, RANDGEN. The idea behind is to perform a few matrix computations on matrices and vectors that hold a lot of data. This is much more efficient than looping over data and performing many scalar computations. RANDGEN subroutine, rather than a DO

loop, is used similarly as RAND function to generate random samples, but it fills an entire matrix at once and stores the simulated data in memory in the x vector.

IMPUTING MISSING DATA USING PROC MI:

Missing values are an issue in a substantial number of clinical analysis studies. Some subjects drop out from the study. Some data are missing due to patient illness or death, invalid measurement, or forgetfulness. A statistical analysis can be biased if incomplete cases are excluded from the analysis due to the intent-to-treat (ITT) principle. The MI procedure performs the MCMC method for missing data imputation, model parameter simulation, and model diagnostics, and to use SAS to perform a Bayesian analysis of data commonly encounter in clinical trials. One may find more details in Scott D Patterson's paper.*[4]* A clinical efficacy data set is used for this example. The data set contains 293 subjects with 3 treatments. The variables are listed as follows:

Variable Name	Description	Valid Value
response	a derived 'change from baseline' variable	numeric value
visit	5 levels of clinical visit	4,6,8,12,16,20
trt	5 levels of treatment including a placebo	1=Dose A, 2=Dose B, 3=Dose C, 4=Dose D, 5=Dose E.
sex	subject's gender	1=male, 2=female
race	4 levels of racial group	1=white, 2=black or African American, 3=Asian, 4=Other
base	standardized baseline value	numeric value
age	standardized age	numeric value
subjid	subject ID	numeric value

Table 1. Description of Sample Clinical Data

Usually PROC MIXED procedure is selected for repeated measures analysis, but SAS MIXED procedure excludes observations with any missing values from the analysis, which will affect the estimates. Suppose that the data are multivariate normally distributed and the missing data are missing at random. Thus PROC MI is necessary for filling arbitrary missing data to obtain more accurate estimates and result.

The following statements invoke the MI procedure and impute missing values using MCMC method.

```
proc mi data=eff out=outmi seed=54321 nimpute=1000;
     mcmc impute=monotone chain=multiple;
     var trt base response;
run;
```

EM (Posterior Mode) Estimates

TYPE	_NAME_	TRT	BASE	RESPONSE
MEAN		3.000000	3.925644	4.790935
COV	TRT	1.998873	0.122785	0.038333
COV	BASE	0.122785	1.150828	0.800170
COV	RESPONSE	0.038333	0.800170	1.624051

Output 1.1 EM (Posterior Mode) Estimates

Above table display the starting mean and covariance estimates used in each imputation. Monte Carlo Simulation (MCMC statement) use existing values as prior information to construct the imputed dataset(outmi). By default, the MI procedure uses the parameter estimates for a posterior mode, that is, the parameter estimates with the highest observed-data posterior density.

In summary SAS MI procedure is very easy to use, only three SAS statements are needed. It provides two optional output datasets, for further analysis.

CONCLUSION

Based on the three simulation/sampling methods we discussed above. DATA step BY-process is most intuitive but time consuming, whereas SAS/IML language is compact and worthy use widely. MI procedure can do missing data imputation meanwhile provide the simulated estimators to have the greatest chance of "success" in a clinical trial.

REFERENCES

[1] Melvin Munsaka. Using SAS for Modeling and Simulation in Drug Development – A Review and Assessment of Some Available Tools, 2011

[2] Peter L. Bonate. Clinical Trial Simulation in Drug Development, 2000

[3] SAS online document: *SIMULATING DATA with SAS* by Rick Wicklin, 2013

[4] Scott D Patterson. SAS Markov Chain Monte Carlo (MCMC) Simulation in Practice, 2007

CONTACT INFORMATION

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

Name: Ye Meng
Enterprise: PPD Inc.
Address: 25F, Raffles Bussiness Center, No. 1 Dongzhimen South Avenue,
Dongcheng District, Beijing
City, State ZIP: Beijing, 100007
Work Phone: +86(0) 10 5763 6125
E-mail: Ye.Meng@ppdi.com
Web: www.ppdi.com