

## A Unified SAS® Macro for Generating Randomisation Schedule

Madan G. Kundu, Ranbaxy Laboratories Limited, Gurgaon, India  
Arjun Roy, Ranbaxy Laboratories Limited, Gurgaon, India

### ABSTRACT

**Randomisation** is the basis and cornerstone for the conduct of the most of clinical trials. Every effort should be made to maintain the element of randomness and reproducibility of the randomisation schedule in a clinical study. In this article, a unified SAS macro (%**RANSCH**) is presented using **PROC PLAN** and **RANUNI** function to generate random permuted block randomisation schedule, choosing **seed number** for each block randomly. This macro is useful to generate randomisation schedule for **parallel**, **cross-over** and **factorial** designs under different trial scenarios such as **varying block sizes**, different **treatment allocation ratios** and presence of **stratification factor(s)**.

### 1. INTRODUCTION

In clinical trials, randomisation helps to avoid bias in subject allocation, tends to produce treatment groups that are similar in known and unknown prognostic factors, and provides a sound statistical basis for the analysis of data. The randomisation schedule of a clinical trial documents the random allocation of treatments to subjects.

Randomisation schedule for a trial can be generated using PROC PLAN available with SAS. Generation of random seed number for PROC PLAN is discussed in section 2. Out of the different types of study design, the most commonly used are parallel, cross-over and factorial designs. These designs vary in terms of balancing of treatments, treatment allocation ratio, presence of stratification factors and also variability of block size. In sections 3 - 5 different designs are discussed under different trial situations (simple or permuted randomisation; fixed or varying block sizes; equal or unequal treatment allocation ratios; stratified or un-stratified) and also the required SAS code to generate randomisation schedule are presented. In section 6, unified SAS macro %RANSCH is discussed. This macro can be used to generate randomisation schedule for parallel, cross-over and factorial designs under different trial scenarios such as varying block sizes, different treatment allocation ratios and presence of various stratification factor(s).

### 2. RANDOM SEED NUMBER GENERATION

*ICH Topic E9 (Statistical Principles for Clinical Trials)* recommends 'The randomisation schedule should be reproducible (if the need arises)'. Using PROC PLAN available in SAS, reproducible randomisation schedule can be generated by specifying seed number. The seed number used for PROC PLAN should be chosen randomly rather than arbitrarily which can be easily achieved using RANUNI function available in SAS dataset. A fragmented SAS code to use random seed number in PROC PLAN is shown below.

```
data _null_;  
x=round(ranuni(0)*1000000);  
call symput ('seed', x);  
run;  
  
title "Seed number = &seed.";  
proc plan seed=&seed.;  
.... /*add the design factors*/  
run;
```

RANUNI function is chosen because random numbers are known to follow uniform distribution.

### 3. PARALLEL DESIGN

The most common clinical trial design is the parallel group design in which subjects are randomised to one of two or more treatment arms. Randomisation for parallel design is affected by different factors such as fixed or varying block sizes, unequal or equal treatment allocation ratio and presence of stratification factor(s).

#### 3.1 SIMPLE RANDOMISATION

It is the most basic method of random treatment assignment. This can be thought of as tossing a coin for each trial participant, A being allocated with "Heads", B with "Tails". Consider a comparative clinical trial with 3 study medications, A: *Test*, B: *Reference* and C: *Placebo*. Suppose the sample size in the study is 300. If the block size is 6 then 50 blocks are required to accommodate 300 subjects. Following program can be used to generate randomisation schedule for this trial.

```
data _null_;  
x=round(ranuni(0)*1000000);  
call symput ('seed', x);title "Seed number = &seed.";  
run;  
  
proc plan seed=&seed.;
```

```

factors block=50 ordered subject=6 ordered treatment=1 of 3 random/noprint;
output out=out
      treatment cvals=('A: Test' 'B: Reference' 'C: Placebo');
run;

proc print data=out noobs;
var block subject treatment;
format subject z3.;
run;

```

The randomisation schedule generated above does not give any assurance that treatments are allocated among the subjects in equal or some other pre-defined treatment allocation ratio in each block. To maintain particular treatment allocation ratio in each block permuted block randomisation is advised.

### 3.2 PERMUTED BLOCK RANDOMISATION

In permuted block randomisation with equal treatment allocation ratio, blocks having equal number of different treatments are used, with the order of treatments within the block being randomly permuted.

Consider the previous example (section 2.1) with three treatments and sample size 300. If the block size is 6 then there are 90 possible permutations in which each of three treatments can appear with equal number of times (i.e. twice). Here randomly chosen permutation (of treatments) are allocated to each block (of subjects) rather than allocating treatments to the subjects. Following SAS code can be used to generate the required randomised schedule.

```

data _null_;
x=round(ranuni(0)*1000000);
call symput ('seed', x);title "Seed number = &seed.";
run;

title "Seed number = &seed.";
proc plan seed=&seed.;
factors block=50 ordered subject=6 ordered/noprint;
treatments treatment=6 random;
output out=out
      treatment cvals=('A: Test' 'A: Test' 'B: Reference' 'B: Reference' 'C: Placebo'
'C: Placebo');
run;

proc print data=out noobs;
var block subject treatment;
format subject z3.;
run;

```

### 3.3 UNEQUAL TREATMENT RANDOMISATION

Sometime due to ethical and other reasons it is required to allocate different treatments in unequal ratio. Consider a trial where *Test*, *Reference* and *Placebo* will be allocated in 2:2:1 ratio and the block size is 10. Then, in each block, *Test* and *Reference* will appear for 4 times each and *Placebo* will appear for 2 times. For this design following code generates randomisation schedule for 300 subjects.

```

data _null_;
x=round(ranuni(0)*1000000);
call symput ('seed', x);title "Seed number = &seed.";
run;

title "Seed number = &seed.";
proc plan seed=&seed.;
factors block=30 ordered subject=10 ordered/noprint;
treatments treatment=10 random;
output out=out
      treatment cvals=('A: Test' 'A: Test' 'A: Test' 'A: Test' 'B: Reference' 'B:
Reference' 'B: Reference' 'B: Reference' 'C: Placebo' 'C: Placebo');
run;

proc print data=out noobs;
var block subject treatment;
format subject z3.;
run;

```

### 3.4 STRATIFIED TREATMENT RANDOMISATION

In any randomised trial it is desirable that the treatment groups should be comparable with regard to those characteristics that might influence the response to the intervention. Stratified block randomisation can restrict chance imbalances to ensure that

the treatment groups are as like as possible for selected prognostic variables or other patient factors. In stratified treatment randomisation, a set of permuted blocks is generated for each combination of prognostic factors.

For example, in a trial of women with breast cancer, it may be important to have similar numbers of pre-menopausal and post-menopausal women in each comparison group. Here the menopausal status can be treated as stratification factor and thus we have two strata. If the total sample size is 300, block size is 10 and treatment allocation ratio of *Test*, *Reference* and *Placebo* is 2:2:1 then we require 15 blocks for each stratum. The following code generates the required randomisation schedule under this scenario.

```
data _null_;
x=round(ranuni(0)*1000000);
call symput ('seed', x);title "Seed number = &seed.";
run;

title "Seed number = &seed.";
proc plan seed=&seed.;
factors strata=2 block=15 ordered subject=10 ordered/noprint;
treatments treatment=10 random;
output out=out
  strata cvals=('Pre-menopausal' 'Post-menopausal')
  treatment cvals=('A: Test' 'A: Test' 'A: Test' 'A: Test' 'B: Reference' 'B: Reference'
'B: Reference' 'B: Reference' 'C: Placebo' 'C: Placebo');
run;

proc print data=out noobs;
var strata block subject treatment;
format subject z3.;
run;
```

### 3.5 RANDOMISATION WITH VARYING BLOCK SIZE

Allocation concealment may be thwarted by an inappropriate choice of randomisation sequence generation. For example, a permuted block design with a fixed block size of four, in an unblinded study where treatment group is revealed at the time of randomisation, may make it easy to predict the next allocation once three patients have been randomised. For this reason, details of block size should not be revealed to the investigator or other study staff. A varying block size technique can also be used (e.g., blocks sizes are chosen randomly from the pre-specified sizes of 4, 6 and 8) for this purpose.

One way for generating randomisation with varying block sizes is to select the block size for each block randomly and then to generate separate randomising schedule for each block. A SAS program to allocate subjects to two treatments in blocks of randomly varying sizes of 4 and 6 can be found in

<http://www.biomedcentral.com/content/supplementary/1471-2288-3-19-S1.doc>

The %RANSCH presented in this paper can be used to generate randomisation schedule with varying block sizes.

## 4. CROSS-OVER DESIGN

In cross over design treatment sequences are allocated to the subjects randomly. Cross-over designs have got advantages over parallel designs like all subjects serve as their own control and error variance is reduced thus reducing sample size needed.

Consider a 3X3 (3-sequence and 3-period) cross over design with three treatments A, B and C. A B C, B C A and C A B are the three treatment sequences. If the block size is 6 and planned number of subject is 48 then 8 blocks are required. The following code generates the desired randomisation schedule for this cross-over design.

```
data _null_;
x=round(ranuni(0)*1000000);
call symput ('seed', x);title "Seed number = &seed.";
run;

title "Seed number = &seed.";
proc plan seed=&seed.;
factors block=8 ordered subject=6 ordered/noprint;
treatments sequence=6 random;
output out=out
  sequence cvals=('A B C' 'A B C' 'B C A' 'B C A' 'C A B' 'C A B');
run;

proc print data=out noobs;
var block subject sequence ;
format subject z3.;
```

```
run;
```

## 5. FACTORIAL DESIGN

In factorial designs two or more treatments are evaluated simultaneously through the use of varying combinations of the treatments. Factorial designs are mainly used to examine the interaction effect between two or more drugs or molecules. They are also used to establish dose-response characteristics and to find out appropriate combination of dose. Randomisation for factorial design can be generated in the similar way as it is done for parallel design. Only difference with parallel design is that in parallel design treatments are randomly allocated whereas in factorial design treatment combinations are randomly allocated to the subjects.

Suppose, if there are 3 dose levels (*A1*, *A2* and *A3*) of study medication *A* and 2 dose levels (*B1* and *B2*) of study medication *B*. As a result, there would be 6 treatment combinations: *A1B1*, *A1B2*, *A2B1*, *A2B2*, *A3B1* and *A3B2*. Also assume that required sample size is the 180. If the block size be the 12 then 15 blocks are required to accommodate these 180 subjects. For this design, the following code generates the desired randomisation schedule.

```
data _null_;
x=round(ranuni(0)*1000000);
call symput ('seed', x);title "Seed number = &seed.";
run;

title "Seed number = &seed.";
proc plan seed=&seed.;
factors block=15 ordered subject=12 ordered /noprint;
treatments treatment=12 random;
output out=out
      treatment cvals=('A1 B1' 'A1 B2' 'A2 B1' 'A2 B2' 'A3 B1' 'A3 B2' 'A1 B1' 'A1 B2'
'A2 B1' 'A2 B2' 'A3 B1' 'A3 B2');
run;

proc print data=out noobs;
var block subject treatment;
format subject z3.;
run;
```

Here it is to be noted that varying block sizes, stratification and unequal treatment allocation may also appear during randomisation of cross-over and factorial study designs.

## 6. MACRO TO GENERATE RANDOMISATION SCHEDULE

In this section a unified SAS macro %*RANSCH* to generate randomisation schedule is described. Flowchart and SAS code of this is presented in the Appendices. This macro is useful to generate randomisation schedule for parallel, cross over and factorial designs under different trial scenarios such as varying block sizes, different treatment allocation ratios and presence of various stratification factor(s).

Requirement of having a SAS macro for generating randomisation schedule is manifold. It is advantageous for the following reasons.

1. No need to write randomisation code in SAS separately for each study. The macro will suffice.
2. One important concern is the validation of code. If there is a single SAS macro then it is to be validated only for once.
3. No need to be/have an SAS expert (at least knowledge in PROC PLAN) to generate randomisation schedule.

### 6.1 FEATURES OF THE MACRO

There are four parts in the macro.

Part 1 verifies whether all the *mandatory* (see section 6.2) macro parameters are defined or not. If not then the macro will stop to run and display an error message along with appropriate suggestion in log window.

Part 2 determines the number of strata, number of subjects per stratum, number of treatments and the number of unique block sizes.

Part 3 checks whether there is enough terms in *RATIO* macro parameter or not. It also checks whether the each block sizes are divisible by the sum of the treatment ratios or not.

Part 4 generates randomisation schedule and print it for each stratum separately. **Within each stratum randomisation schedules are generated for each block separately using randomly chosen seed number.**

### 6.2 INPUT OF THE MACRO

There are following macro parameters in %*RANSCH* that need to be specified.

1. TRTNAME
2. RATIO

3. BLKSIZE
4. STRATANAME
5. NUMSUBJ
6. STUDYTITLE

STRATANAME is OPTIONAL whereas the remaining parameters are MANDATORY to specify.

**TRTNAME:**

User needs to specify here the treatment names separated by backslash ('/'). For example, if there are three study medications such as *Test*, *Placebo* and *Reference* drug then macro parameter specification for TRTNAME will be:

```
trtname=Test drug/Placebo/Reference drug
```

For the cross-over design this should be the name of treatment sequences. For a 3X3 cross over design TRTNAME can be specified as follows:

```
trtname=A B C/B C A/C A B
```

**RATIO:**

Here treatment allocation ratio should be specified. If the treatment allocation ratio for *Test*, *Reference* and *Placebo* is 2:2:1 ratio then macro parameter specification for TRTNAME and RATIO should be:

```
trtname=Test /Reference /Placebo
ratio=2 2 1
```

For the cross over design this would be the ratio of sequences. If the number of terms in RATIO macro parameter is less than the number of treatments (or sequences) specified in TRTNAME then the macro will stop and show following error message.

```
ERROR: Number of elements in RATIO macro parameter is less than the number of treatments
```

If the number of terms in RATIO macro parameter is more than the number of treatments (or sequences) then the additional terms from right will be simply ignored.

**BLKSIZE:**

Here user needs to specify the block size(s). In case of fixed block size this will be single value. For example if the block size is 10 then parameter specification will be;

```
blksize=10
```

If there is more than one block size (in case of varying block size) set of block sizes should be specified separated by space. For example if there is two block sizes 10 and 15, parameter specification will be:

```
blksize=10 15
```

Block sizes should be divisible by the sum of the treatment-ratios; otherwise the macro will stop and display the Error message in the log window. For example if the block size is 10 and the treatment ratio is 1:1:1 then the following error message will be displayed.

```
ERROR: Block size 10 is not divisible by sum of the treatment ratios (3)
```

**STRATANAME:**

Name of the strata should be specified for this macro parameter. This parameter is optional and the default number of strata is 1.

For example, if the stratification factor is *Gender* then the strata will be *Male* and *Female*. In that case this macro parameter should be specified in the following way:

```
strataname=Male/Female
```

**NUMSUBJ:**

Number of subjects to be randomised. If the number of subjects is 300 then specification for this parameter will be:

```
numsubj=300
```

**STUDYTITLE:**

Study title should be specified.

**6.3 USE OF %RANSCH: ILLUSTRATION**

**Case1: Three treatment arms parallel design, equal treatment allocation ratio, unstratified and fixed block size**

To generate the randomisation schedule for the design discussed in section 3.2, %RANSCH can be used in the following way:

```
%ransch ( trtname=A: Test/B: Reference/C: Placebo,
          ratio=1 1 1,
          blksize=6,
          numsubj=300,
          studytitle=XYZ study )
```

**Case2: Three treatment arms parallel design, unequal treatment allocation ratio, unstratified and fixed block size**

To generate the randomisation schedule for the design discussed in section 3.3, %RANSCH can be used in the following way:

```
%ransch ( trtname=A: Test/B: Reference/C: Placebo,
           ratio=2 2 1,
           blksize=10,
           numsubj=300,
           studytitle=XYZ study )
```

**Case3: Three treatment arms parallel design, unequal treatment allocation ratio, stratified and fixed block size**

To generate the randomisation schedule for the design discussed in section 3.4, %RANSCH can be used in the following way:

```
%ransch ( trtname=A: Test/B: Reference/C: Placebo,
           ratio=2 2 1,
           blksize=10,
           strataname=Pre-menopausal/Post-menopausal,
           numsubj=300,
           studytitle=XYZ study )
```

**Case4: Three treatment arms parallel design, unequal treatment allocation ratio, stratified and varying block size**

To generate the randomisation schedule with varying block sizes of 10 and 15 and keeping other things as same as the design described in the section 3.4, %RANSCH can be used in the following way:

```
%ransch ( trtname=A: Test/B: Reference/C: Placebo,
           ratio=2 2 1,
           blksize=10 15,
           strataname=Pre-menopausal/Post-menopausal,
           numsubj=300,
           studytitle=XYZ study )
```

**Case5: Three treatments, three sequences and three period cross-over design, equal sequence allocation ratio, unstratified and fixed block size**

To generate the randomisation schedule for the cross over design discussed in section 4, %RANSCH can be used in the following way:

```
%ransch ( trtname=A B C/B C A/C A B,
           ratio=1 1 1,
           blksize=6,
           numsubj=48,
           studytitle=XYZ study )
```

**Case6: Factorial design with three levels of one factor and two levels of another factor, equal allocation ratio, unstratified and fixed block size**

To generate the randomisation schedule for the factorial design discussed in section 5, %RANSCH can be used in the following way:

```
%ransch ( trtname=A1 B1/A1 B2/A2 B1/A2 B2/A3 B1/A3 B2,
           ratio=1 1 1 1 1 1,
           blksize=12,
           numsubj=180,
           studytitle=XYZ study )
```

**6.4 OUTPUT OF %RANSCH**

Output of %RANSCH contains the randomisation schedule which includes subject id and the allocated treatment. The randomisation schedule for each block is generated in different page(s). Randomisation schedule for first two blocks generated on invoking %RANSCH (case1, section 6.3) looks as below:

Page1:

| Subject<br>id                     | Treatment    |
|-----------------------------------|--------------|
| Block 1: Size=6, Seed number:8055 |              |
| 001                               | C: Placebo   |
| 002                               | A: Test      |
| 003                               | C: Placebo   |
| 004                               | B: Reference |
| 005                               | A: Test      |
| 006                               | B: Reference |

Page2:

| Subject<br>id                     | Treatment    |
|-----------------------------------|--------------|
| Block 2: Size=6, Seed number:3604 |              |
| 001                               | B: Reference |
| 002                               | A: Test      |

|     |              |
|-----|--------------|
| 003 | C: Placebo   |
| 004 | B: Reference |
| 005 | C: Placebo   |
| 006 | A: Test      |

## 7. CONCLUSION

Using the %RANSCH presented in this paper randomisation schedule for Parallel, Cross-over and Factorial design under different trial situation (e.g., varying block size, unequal treatment allocation ratio and presence of stratification factors) can be generated. Unified SAS macro approach is advantageous in terms of saving time and validation perspectives. This approach of using SAS macro can also be extended to generate randomisation schedule for other designs like William's design, Balaam's design, Balanced Incomplete Block (BIB) design etc.

## 8. REFERENCE

1. Altman, D.G. and Bland, J.M. (1999), "Treatment allocation in controlled trials: why randomize?", *BMJ*, **318**: 1209.
2. Altman, D.G. and Bland, J.M. (1999). "How to randomize?", *BMJ*, **319**: 703-704.
3. Beller E.M., Gebski V. and Keech A. C. (2002), "Randomisation in clinical trial", *MJA*, **77**: 565-567
4. Chow, S.C and Liu, J.P. (2004), Design and analysis of clinical trials and bioequivalence studies, *Wiley Interscience*.
5. Deng C. and Graz J. (2002), "Generating randomization schedules using SAS<sup>®</sup> Programming", *SUGI*, **27**: Paper 267
6. ICH Topic E9: Statistical Principles for Clinical Trials, 1998, Human Medicines Evaluation Unit, *The European Agency for the Evaluation of Medicinal Products*.
7. SAS institute Inc. (1999), SAS/Stat User's Guide, Version 8, Cary, NC: *SAS Institute Inc*.

## ACKNOWLEDGEMENT

The authors are highly thankful to Dr. Gowrishankar and Dr. Jaythirtha M G for their useful suggestions.

## CONTACT INFORMATION

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

Madan Gopal Kundu

Biostatistician & SAS Programmer, CDM, MACR, Ranbaxy Laboratories Limited

Plot no 77B, Sector 18, Gurgaon, Haryana, India, 122001

Work Phone: 91-124-5194113, Mobile: +919868788406

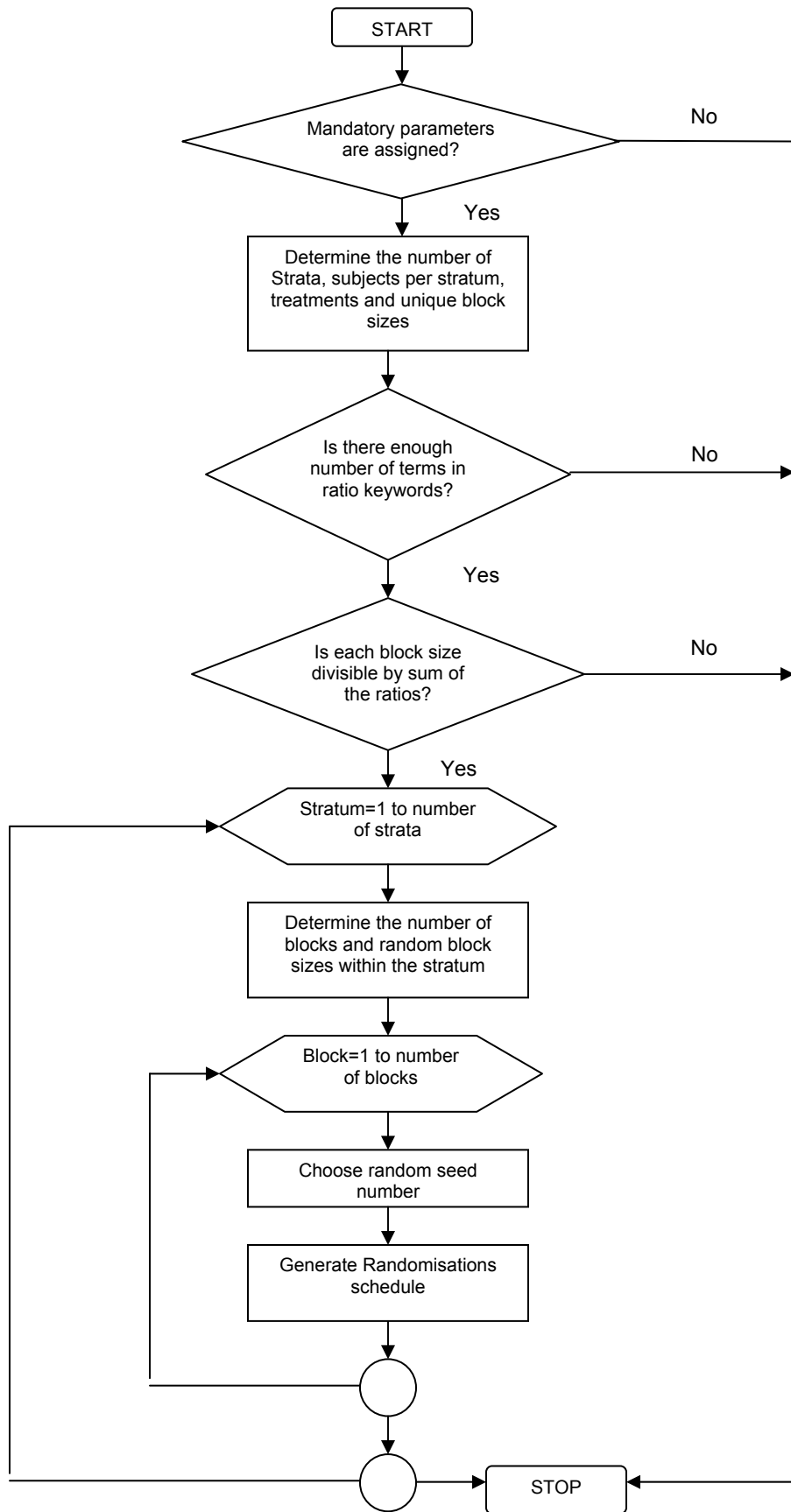
E-mail: [Madan.Kundu@ranbaxy.com](mailto:Madan.Kundu@ranbaxy.com); [madan4331@yahoo.co.in](mailto:madan4331@yahoo.co.in)

Web: <http://www.freewebs.com/madanstata/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

APPENDIX – I: Flowchart for %RANSCH



## APPENDIX – II: SAS Code for %RANSCH

```
/* ***** */
/* PURPOSE:      To generate random permuted block randomisation schedule for      */
/*               Parallel, Cross-over and Factorial designs with varying block    */
/*               sizes, different treatment allocation ratio and stratification    */
/*               factor                                                            */
/* AUTHOR:       Madan Gopal Kundu (mail: Madan.Kundu@Ranbaxy.com)                */
/* DATE:         28-Nov-2006                                                       */
/* MANDATORY                                          */
/* PARAMETERS:   trtname - Treatment or sequence names separated by backslash '/' */
/*               ratio - Treatment or sequence allocation ratio, separated by space */
/*               blksize - Block size. More than one block sizes should be separated*/
/*               by space                                                            */
/*               numsubjj - Number of subjects                                    */
/*               studytitle- Title of the study                                    */
/* OPTIONAL                                          */
/* PARAMETERS:   strataname - Name of the strata separated by backslash '/'      */
/* MACRO VARIABLES                                          */
/* USED INSIDE THE                                          */
/* MACRO:        numtrata - Number of strata                                       */
/*               subj - Number of subjects in each stratum                         */
/*               numtrt - Number of treatments                                     */
/*               numblksize - Number of unique block sizes                         */
/*               rtotal - Sum of the ratios                                        */
/*               ratio1, ratio2, ... - Individual ratios                           */
/*               blknum - Number of blocks within astratum                         */
/*               blk1, blk2, ... - Individual Block sizes                          */
/*               seed - random seed value to generate                             */
/*               randomization schedule                                           */
/* ***** */
%macro ransch (trtname=, ratio=, blksize=, strataname=, numsubjj=, studytitle= );
/*----- PART - I -----*/
/*CHECKING WHETHER ALL THE MANDATORY MACRO-PARAMETERS ARE DEFINED OR NOT*/
%if %length(&trtname.)=0 %then
  %do;
    %put ERROR: TREATMENT NAMES are not specified in TRTNAME macro parameter position;
    %put SUGGESTION: Specify the treatment name separated by separated by Backslash '/';
    %goto finish;
  %end;

%if %length(&ratio.)=0 %then
  %do;
    %put ERROR: TREATMENT RATIO are not specified in RATIO macro parameter position;
    %put SUGGESTION: Specify the treatment ratio separated by separated by space;
    %goto finish;
  %end;

%if %length(&blksize.)=0 %then
  %do;
    %put ERROR: BLOCK SIZE(s) are not specified in BLKSIZE macro parameter position;
    %put SUGGESTION: Specify the block sizes separated by separated by space;
    %goto finish;
  %end;

%if %length(&numsubjj.)=0 %then
  %do;
    %put ERROR: NUMBER OF SUBJECTS are not specified in NUMSUBJJ macro parameter position;
    %put SUGGESTION: Specify the number of subjects;
    %goto finish;
  %end;

%if %length(&studytitle.)=0 %then
  %do;
    %put ERROR: STUDY TITLE are not specified in STUDYTITLE macro parameter position;
    %put SUGGESTION: Specify the study title;
  %end;

```

```

    %goto finish;
%end;

/*----- PART - II -----*/
/*DETERMINING THE NUMBER OF STRATA*/
%let numstrata=1; /*Default number of strata*/
%put &numstrata.;
%if %length(&strataname.)>0 %then %do;
    %let s=1;
    %let stratum=%scan(&strataname, 1, '/');
    %do %while (%length(&stratum.)>0);
        %let stratum=%scan(&strataname, &s., '/');
        %let s=%eval(&s.+1);
    %end;
    %let numstrata=%eval(&s.-1);
%end;

/*DETERMINING NUMBER OF SUBJECTS PER STRATUM*/
%let subj=%eval(&numsubj./&numstrata.);

/*DETERMINING THE NUMBER OF TREATMENTS*/
%let v=1;
%let trt=%scan(&trtname., 1, %str('/'));
%do %while (%length(&trt.)>0);
    %let v=%eval(&v.+1);
    %let trt=%scan(&trtname, &v., '/');
%end;
%let numtrt=%eval(&v.-1);

/*DETERMINING THE NUMBER OF UNIQUE BLOCK SIZES*/
%let r=1;
%let k=%scan(&blksize., &r., ' ');
%do %while (%length(&k.)>0);
    %let r=%eval(&r.+1);
    %let k=%scan(&blksize., &r., ' ');
%end;

%let numblksize=%eval(&r.-1);

/*----- PART - III -----*/
/*CHECKING WHETHER THERE IS ENOUGH TERMS IN RATIO KEY WORDS*/
%let rcheck=%scan(&ratio., &numtrt.);

%if %datatype(&rcheck.)=CHAR %then
    %do;
        %put ERROR: Number of elements in RATIO keyword is less than the number of treatments;
        %put SUGGESTION: Correct this problem in RATIO macro parameter position of the macro;
        %goto finish;
    %end;

/*TAKING EACH TERMS OF RATIO IN SEPARATE MACRO-VARIABLES AND ALSO COMPUTING SUM OF THE RATIOS*/
%let rttotal=0;
%do v=1 %to &numtrt.;
    %let ratio&v.=%scan(&ratio., &v., ' ');
    %let rttotal=%eval(&rttotal.+&&ratio&v.);
%end;

/*CHECKING WHETHER EACH BLOCK SIZES ARE DIVISIBLE BY THE SUM OF THE RATIOS OR NOT*/
%do r=1 %to &numblksize.;
    %let k=%scan(&blksize., &r., ' ');
    %let Div1=%sysevalf(&k./&rttotal.);
    %let Div2=%eval(&k./&rttotal.);
    %if &Div1.>&Div2. %then
        %do;
            %put ERROR: Block size &k. is not divisible by sum of the treatment ratios
            (&rttotal.) ;
        %end;
    %end;
%end;

```

```

        %put SUGGESTION: Choose block size(s) as multiple of sum of the treatment ratios
(&rtotal.) ;
        %goto finish;
    %end;
%end;

/*----- PART - IV -----*/
/*GENERATING THE RANDOMISING SCHEDULE FOR EACH STRATUM*/

%do stratum=1 %to &numstrata.;

    /*DETERMINING NUMBER OF BLOCKS AND BLOCK SIZES RANDOMLY FOR EACH BLOCKS*/

    data blksize_&stratum.;
    array blksize(*) blksize1 - blksize&numblksize.;
    %do i=1 %to &numblksize.;
    blksize&i.=%scan(&blksize.,&i., ' ');
    %end;
    numsubj=&subj.;
    total=0;
    do while (total<numsubj);
        rand=ranuni(0)*100000;
        check=input(rand,5.0);
        determ=mod(check,&numblksize.);
        %do i=1 %to &numblksize.;
            if determ=%eval(&i.-1) then size=blksize&i.;
        %end;
        total=total+size;
        if total>numsubj then do;
            diff=total-numsubj;
            size=size-diff;
            total=total-diff;
        end;

        output;
    end;
run;

    /*ASSIGNING THE BLOCK SIZES AND BLOCK NUMBERS IN DIFFERENT MACRO VARIABLES*/
    data blksize_&stratum.;
    set blksize_&stratum. end=eof;
    call symput ('blk' || left(_n_), input(compress(size),3.0));
    if eof=1 then call symput ('blknum', _n_);
run;

    /*GENERATING THE RANDOMISATION SCHEDULE FOR EACH BLOCKS SEPARATELY*/
    %do b=1 %to &blknum.;

        /*DETERMINING REPEATATION OF TREATMENTS IN THE BLOCK AS PER TREATMENT
ALLOCATION RATIO*/
        data _null_;
        blksize=&&blk&b.;
        rtotal=&rtotal;
        %do v=1 %to &numtrt.;
            ratio=&&ratio&v.;
            size=ratio*blksize/rtotal;
            call symput ('size' || left(&v.), input(compress(size),3.));
            output;
        %end;
        run;

        /*GENERATING RANDOM SEED*/
        data _null_;
        seed=int(ranuni(0)*10000);
        call symput ('seed', seed);
        run;

        /*GENERATING RANDOMISATION SCHEDULE */

```

```

proc plan seed=&seed.;
factors subj=&&blk&b.. ordered /noprint;
treatments treat=&&blk&b.. random;
output out=Block&b._&stratum.
      treat nvals=( %do i=1 %to &numtrt.;
                   %do j=1 %to %eval(&&size&i.);
                   &i.
                   %end;
                   %end; ) ;

run;
quit;

/*GENERATING TITLE FOR EACH BLOCK*/
data Block&b._&stratum.;
set Block&b._&stratum.;
block=&b.;
seed=&seed.;
size=&&blk&b..;
length blktitle $ 70;
blktitle='Block ' || compress(block) || ': Size=' || compress(size) || ', Seed
number:' || compress(seed);
run;

%end;

/*MERGING ALL THE BLOCK RANDOMISATION SCHEDULES GENERATED FOR THIS STRATUM*/
data Strata&stratum.;
merge %do b=1 %to &blknum.;
      Block&b._&stratum.
      %end;;
by block;
strata=&stratum.;
run;

/*FORMATTING TREATMENTS*/
proc format;
value trt
      %do i=1 %to &numtrt.;
      %let trt=%scan(&trtname.,&i.,%str('/'));
      &i.="&trt."
      %end;
run;

/*PRINTING OF RANDOMISATION SCHEDULE*/
proc report data=Strata&stratum. split='*' nofs;
columns block blktitle subj treat ;
define block/order noprint;
define blktitle/order noprint;
define subj/display 'Subject*id' format=z3. center;
define treat/display 'Treatment' format=trt. center;
break after block/page;
compute before blktitle;
line blktitle $70.;
endcomp;
title1 font=timesroman height=1.2 justify=center "Study Title: &studytitle.";
title2 font=timesroman height=1.2 justify=center "Number of
Treatments/Sequence=&numtrt.";
title3 font=timesroman height=1.2 justify=center "Treatment/Sequence allocation
ratio=&ratio.";
title4 font=timesroman height=1.2 justify=center "Number of subjects =&numsubj.";
%if %length(&strataname.)>0 %then %do;
      title6 font=timesroman height=1.2 "Stratum=&stratum.";
      %end;
run;
%end;
%finish;
%mend;

```