PharmaSUG 2021 - Paper SA-138

Willing to Play with William's Design: An Exemplification of Randomization Schedule Generation using SAS

Abhyuday Chanda, Quartesian Research Private Limited

ABSTRACT

Crossover design is one of the most used design in the clinical trials. With 'n' treatments in a 'n' period crossover design, an imbalance of carryover effect of 1st order may be observed. To maintain the balance, one needs to make sure the number of times a treatment (Drug A) preceding another treatment (Drug B) is same as that treatment (Drug A) succeeding the other treatment (Drug B). A Williams design, which is a (generalized) Latin Square Design (LSD), balanced for first order carryover effects is applicable in these cases. If the number of treatments to be tested is even, the design is a Latin square (nXn), otherwise it consists of two Latin squares (2nXn). This paper will demonstrate the readers the step-by-step process of generating randomization schedules for both even (6X6) and odd (3X3) number of treatments with a Williams design using SAS®. The brief steps for generating the same are: 1) Generate a Latin square; 2) Create its mirror image and augment that to the main Latin square; 3) Interlace the columns of the main and the mirror Latin square; 4) Create the final sequences based on number of treatments; and 5.) Use those sequences for generating the final randomization schedule.

Key words: Carryover effect, Crossover design, Latin square design, Randomization Schedule, William's design

INTRODUCTION

In clinical trials, the most widely used crossover design is an AB/BA, i.e. a 2 treatment 2 period study. In case of n treatment n period design , where n> 2, along with the equal occurrence of each treatment in each period, the 1st order carryover effect can be balanced using William's design. This paper will discuss with examples how a randomization schedule using William's design can be created for a study with n number of treatment, n period and a*n subjects, where a is an integer and ≥ 1 using SAS.

CREATING A RANDOMIZATION SCHEDULE FOR WILLIAM'S DESIGN

An nXn William square must satisfy the following 3 assumptions:

1. Each treatment is given only once to each patient.
2. Each treatment occurs the same number of times in each period (once if n is even, twice if n is odd)
3. The number of times a treatment (say Drug A) precedes another treatment (say Drug B) should be same as that treatment (Drug A) succeeds the other treatment (Drug B)

Table 1: An example of 3X3 Latin Square Design (LSD)

Sequence	Period 1	Period 2	Period 3
1	A	B	C
2	B	C	A
3	C	A	B

Table 1 presents an example of 3X3 Latin Square Design(LSD). In the Table 1, although the assumptions 1 and 2 are satisfied, however the design fails to satisfy assumption 3. For the drug A and B, it is observed that A precedes B twice (sequence 1 and 3), but B never precedes A. Thus it doesn't satisfy assumption 3. Thus this design does not qualify to be a William's Design.

Table 2: An example of William's design for 3 treatment 3 period

Sequence	Period 1	Period 2	Period 3
1	A	B	C
2	B	C	A
3	C	A	B
4	B	A	C
5	A	C	B
6	C	B	A

In Table 2 All the 3 assumptions are met. For example B succeeding A is occurring twice in sequence 1 and 3, while A succeeding B is also occurring twice in sequence 4 and 6.

CASE 1: ODD NUMBER OF TREATMENTS

This section discusses the steps to generate a randomization schedule for an n treatment, n period, 2*n sequence with a*n subjects using William's design, when n is odd.

Step 1: Create a format for treatment and generate 2 seed values:

```
proc format;
value trt
      1="A"
      2="B"
      3="C"
;
run;


data _null_;
        seed00=int(datepart(datetime()));
        seed11=int(timepart(datetime()));
        call symputx("seed",seed00);
        call symputx("seed1",seed11);
run;
```

Step 2: Generate the standard LSD and apply the format 'trt':

```
proc plan seed=&seed;
      factors period=3  seq=3  ordered/ noprint;
      treatments tt=3 cyclic ;
      output out=ggx1 ;
quit;
```

The dataset looks like as below Table 3

Table 3: Dataset GGX1 for odd n=3

PERIOD	SEQ	TT
2	1	1
2	2	2
2	3	3
3	1	2
3	2	3
3	3	1
1	1	3
1	2	1
1	3	2

```
data gg     ;
      set ggx1    ;
      tmts=put(tt, trt.);
run;
proc sort data=gg;
      by seq period;
run;
```

The dataset looks like as below Table 4

Table 4: Dataset GG for odd n=3

PERIOD	SEQ	TT	TMTS
1	1	3	C
2	1	1	A
3	1	2	B
1	2	1	A
2	2	2	B
3	2	3	C
1	3	2	B
2	3	3	C
3	3	1	A

Step 3: Transpose the dataset 'GG' as follows:

```
proc transpose data=gg out=gg1(drop=_name_) prefix=period;
      id period;
      by seq;
      var tmts;
run;
```

The dataset looks like as below Table 5

Table 5: Dataset GG1 for odd n=3

SEQ	PERIOD1	PERIOD2	PERIOD3
1	C	A	B
2	A	B	C
3	B	C	A

Step 4: Obtain a mirror image of the standard LSD;

```
data gg2;
      set gg1;
      period11=period3;
      period22=period2;
      period33=period1;
run;
```

The dataset looks like as below Table 6

Table 6: Dataset GG2 for odd n=3

SEQ	PERIOD1	PERIOD2	PERIOD3	PERIOD11	PERIOD22	PERIOD33
1	C	A	B	B	A	C
2	A	B	C	C	B	A
3	B	C	A	A	C	B

Step 5: Interlace each column of the standard LSD and its mirror image

```
proc sql ; create table gg21 as select seq,
      period1 as p1, period11 as p2, period2 as p3,
      period22 as q1, period3 as q2, period33 as q3
      from gg2;
quit;
```

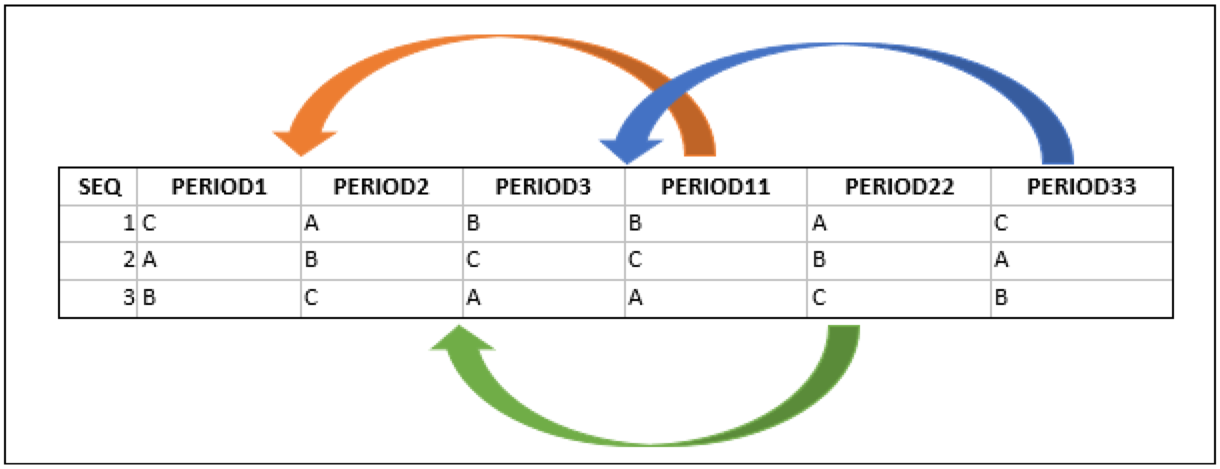
The dataset looks like as below Table 7

Table 7: Dataset GG21 for odd n=3

SEQ	P1	P2	P3	Q1	Q2	Q3
1	C	B	A	A	B	C
2	A	C	B	B	C	A
3	B	A	C	C	A	B

To understand interlace please see the below Figure 1:

Figure 1: Interlacing the actual and the Mirror image of the standard Latin square



SEQ	PERIOD1	PERIOD2	PERIOD3	PERIOD11	PERIOD22	PERIOD33
1	C	A	B	B	A	C
2	A	B	C	C	B	A
3	B	C	A	A	C	B

Here the variables are rearranged, and Period x and Period xx are kept consecutively(Period1, Period11, Period2, Period22, Period3 and Period33). Post that, the first 3 variables (Period1, Period11 and Period2) are renamed as P1,P2 and P3 (Latin square 1) and the last 3 variables (Period22, Period3 and Period33) are renamed as Q1, Q2 and Q3 (Latin square 2).

Step 6: Creating the William's design:

```
data gg22;
      set gg21(in=a) gg21(in=b);
      treatseq=_n_;
      if a then do; sequence=catx("-",p1,p2,p3);end;
      if b then do; sequence=catx("-",q1,q2,q3);end;
      keep treatseq sequence;
run;
```

The dataset looks like as below Table 8

Table 8: Dataset GG22 for odd n=3

TREATSEQ	SEQUENCE
1	C-B-A
2	A-C-B
3	B-A-C
4	A-B-C
5	B-C-A
6	C-A-B

The final sequences of the design can be obtained by concatenating Px variables and Qx variables separately and then appending them as described in the above code.

Step 7: Creating the randomization schedule using the William's design matrix:

- Create a schedule using proc plan as follows:

```
proc plan seed=&seed1;
```

```
      factors blocks=6 treatseq=6 random/noprint;
      output out=rand_n treatseq nvals=(1 2 3 4 5 6) random;
      run;
quit;
```

- Create a dummy variable x as _n_ in rand_n dataset

```
data rand_n ;
      set rand_n  ;
      x=_n_;
run;
```

- Sort and merge rand_n and gg22 using treatseq

```
proc sort data=rand_n   ;
      by treatseq ;
run;

proc sort data=gg22     ;
      by treatseq ;
run;

data final_3      ;
      merge gg22  (in=a) rand_n (in=b);
      by treatseq ;
run;
```

- Sort the merged dataset by blocks

```
proc sort data=final_3  ;
      by blocks   ;
run;
```

- Assign random number variable ‘Randnum’ as 1000+x (dummy variable created above)

```
data final_odd    ;
      set final_3 ;
      randnum=1000+x;
      keep sequence randnum;
run;
```

- Sort the ‘final_odd’ dataset by Randnum

```
proc sort data=   final_odd;
      by randnum  ;
run;
```

The final randomization schedule looks like as below Table 9

Table 9: A 3 treatment, 3 period 6 sequence randomization schedule for 36 subjects

RANDNUM	SEQUENCE
1001	A-B-C
1002	C-A-B
1003	B-C-A
1004	C-B-A
1005	B-A-C
1006	A-C-B
1007	B-A-C
1008	C-A-B
1009	A-C-B
1010	B-C-A
1011	A-B-C
1012	C-B-A
1013	B-C-A
1014	B-A-C
1015	C-B-A
1016	A-C-B
1017	C-A-B
1018	A-B-C
1019	B-C-A
1020	B-A-C
1021	C-B-A
1022	A-C-B
1023	C-A-B
1024	A-B-C
1025	B-C-A
1026	A-C-B
1027	C-A-B
1028	B-A-C
1029	C-B-A
1030	A-B-C
1031	C-A-B
1032	C-B-A
1033	A-C-B
1034	B-A-C
1035	A-B-C
1036	B-C-A

CASE 2: EVEN NUMBER OF TREATMENTS

This section discusses the steps to generate a randomization schedule for an n treatment, n period, n sequence with a*n subjects using William's design, when n is even.

Follow steps 1 to 5 from the case 1.

Step 6: Creating the William's design

Any one of the two Latin squares created in step 5 can be used. To maintain the unbiasedness the following approach can be used:

```
data _null_;
   ls = rand("Integer", 1, 2);
   output;
   call symputx("ls",ls);
run;

data gg22   ;
      set gg21    ;
      if &ls=1 then do; sequence=catx("-",p1,p2,p3,p4,p5,p6);end;
      else do ; sequence=catx("-",q1,q2,q3,q4,q5,q6);end;
run;
```

Step 7 to be followed in the same way as in odd 'n'

The full code for 'even n' is given below

```
/* "RANDOMIZATION SCHEDULE FOR A 6X6 6 seq WILLIAMS DESIGN";*/

proc format;
value trt
      1="B"
      2="C"
      3="D"
      4="E"
      5="F"
      6="G"
;
run;


data _null_;
        seed00=int(datepart(datetime()));
        seed1=int(timepart(datetime()));
        call symputx("seed",seed00);
        call symputx("seed1",seed1);
run;
*** Generating the standard Latin Square;
proc plan seed=&seed;
      factors period=6 ordered seq=6 ordered/ noprint;
      treatments tt=6 cyclic;
      output out=ggx ;
quit;


data gg     ;
      set ggx     ;
      tmts=put(tt,trt.);
run;

proc sort data=gg;
      by seq period;
run;

proc transpose data=gg out=gg1(rename=(col1=period1 col2=period2 col3=period3
col4=period4 col5=period5 col6=period6
 ) drop=_name_);
      by seq;
      var tmts;
run;

*** Obtain a mirror image of the standard Latin square;
data gg2;
      set gg1;
      period11=period6;
      period22=period5;
      period33=period4;
      period44=period3;
      period55=period2;
      period66=period1;
run;

*** Interlace each column of the standard and its mirror image;
proc sql ;
create table gg21 as
      select seq,period1 as p1,period11 as p2 ,period2 as p3,period22 as
      p4,period3 as p5,period33 as p6,

      period4 as q1,period44 as q2,period5 as q3 ,period55 as q4,period6 as
      q5,period66 as q6

from gg2
;
quit;

/*Random selection of 1 Latin square from the available 2*/
data _null_;
   ls = rand("Integer", 1, 2);
   output;
   call symputx("ls",ls);
run;

data gg22   ;
      set gg21    ;
      if &ls=1 then do; sequence=catx("-",p1,p2,p3,p4,p5,p6);end;
      else do ; sequence=catx("-",q1,q2,q3,q4,q5,q6);end;
      treatseq=_n_;
run;

/*****Creating the full schedule for 36 subjects***********************/
proc plan seed=&seed1;
      factors blocks=6 treatseq=6 random/noprint;
      output out=rand_n treatseq nvals=(1 2 3 4 5 6) random;
run;
```

```
quit;

data rand_n ;
set rand_n  ;
x=_n_;
run;

proc sort data=rand_n   ;
      by treatseq ;
run;

proc sort data=gg22     ;
      by treatseq ;
run;

data final_6      ;
      merge gg22  (in=a) rand_n (in=b);
      by treatseq ;
run;

proc sort data=final_6  ;
      by blocks   ;
run;

data final_even   ;
      set final_6 ;
      randnum=1000+x;
      keep sequence randnum;
run;

proc sort data=   final_even;
      by randnum  ;
run;
```

The final randomization schedule looks like as below Table 10:

Table 10: A 6 period 6 treatment 6 sequence randomization schedule for 36 subjects

RANDNUM	SEQUENCE
1001	E-D-F-C-G-B
1002	F-E-G-D-B-C
1003	B-G-C-F-D-E
1004	D-C-E-B-F-G
1005	C-B-D-G-E-F
1006	G-F-B-E-C-D
1007	F-E-G-D-B-C
1008	G-F-B-E-C-D
1009	E-D-F-C-G-B
1010	B-G-C-F-D-E
1011	D-C-E-B-F-G
1012	C-B-D-G-E-F

1013	F-E-G-D-B-C
1014	D-C-E-B-F-G
1015	G-F-B-E-C-D
1016	C-B-D-G-E-F
1017	B-G-C-F-D-E
1018	E-D-F-C-G-B
1019	G-F-B-E-C-D
1020	D-C-E-B-F-G
1021	C-B-D-G-E-F
1022	B-G-C-F-D-E
1023	F-E-G-D-B-C
1024	E-D-F-C-G-B
1025	D-C-E-B-F-G
1026	G-F-B-E-C-D
1027	C-B-D-G-E-F
1028	B-G-C-F-D-E
1029	E-D-F-C-G-B
1030	F-E-G-D-B-C
1031	D-C-E-B-F-G
1032	G-F-B-E-C-D
1033	E-D-F-C-G-B
1034	F-E-G-D-B-C
1035	C-B-D-G-E-F
1036	B-G-C-F-D-E

CONCLUSION

The paper describes in detail the stepwise process of creating randomization schedule in SAS® for William's design using real time examples for both the cases when n is odd and even. The outputs presented here may differ from user to user as the seed values are not kept fixed.

REFERENCES

Deng, Chunqin and Graz, Julia. 2002. "Generating Randomization Schedules Using SAS® Programming." SUGI 27 , Paper 267.

Wang, Bing-Shun ; Wang, Xiao-Jin and Gong, Li-Kun. February 2009. "The Construction of a Williams Design and Randomization in Cross-Over Clinical Trials Using SAS" Journal of Statistical Software, Volume 29, Code Snippet 1.

ACKNOWLEDGMENTS

The author is grateful to Sakthivel Sivam, PhD and Sakshi Kaushik, PhD for their review comments and feedbacks.

CONTACT INFORMATION

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

Abhyuday Chanda
Senior Biostatistician, Biostatistics,
Quartesian Research Private Limited
E-mail: abhyuday.chanda@quartesian.com
abhyuday.chanda@yahoo.com