

## Minimization: Testing A Dynamic Randomization Algorithm

Donna E. Levy & Emily A. Blood, Dana-Farber Cancer Institute

### Abstract:

Randomization and stratification are necessary components of enrolling patients onto a clinical trial with more than one treatment. The purpose is to eliminate treatment selection bias and ensure balance of prognostic factors across treatment arms.

There are several randomization algorithms, dynamic and non-dynamic, designed to balance treatment assignments amongst patients with respect to prognostic factors. However, non-dynamic methods can result in overall treatment imbalance.

Minimization is a dynamic randomization algorithm designed to minimize imbalance between treatments taking stratification factors into account. Based on characteristics of the current patient and treatment assignments and characteristics of enrolled patients an imbalance score is computed. The treatment with the lowest imbalance score is assigned.

SAS<sup>®</sup> code was written to test newly adopted randomization algorithms utilizing minimization. The algorithms differ in the method of selecting treatment as well as the measure used to assess imbalance.

The code is flexible and accommodates specifications such as number of treatments, number of prognostic factors, number of levels within a prognostic factor, treatment allocation proportions and importance of each prognostic factor. It accommodates different measures of imbalance and methods of treatment selection.

The code includes MACROS, MACRO variables, DO loops, data MERGEs, and several forms of data manipulations.

### Introduction:

Randomization is a necessary component of most multi-arm clinical trials. There are several objectives of randomization such as eliminating personal selection bias as well as other biases that may affect the assessment of the results of the trial (Everitt, 1994). Randomization attempts to balance measured as well as unobserved concomitant variables between the treatment groups. When treatments are assigned randomly the group of patients on each arm of the trial should not differ with respect to distribution of known and unknown factors that may influence patient outcome (Simon, 1985). If however, there are factors that are known to influence patient outcome to treatment these should be taken into account when randomizing patients. For example, if older patients have a worse outcome on a particular experimental treatment in comparison to younger patients and all the older patients were assigned to the experimental treatment arm. If the experimental treatment is an overall better treatment, the superiority of the treatment would probably not be shown in this situation. We therefore want to balance the treatment arms with respect to known prognostic factors. These factors are called stratification factors when used in the randomization process.

There are several different randomization algorithms. These can be classified as either static or dynamic. Static algorithms assign treatment in a sequence established prior to any patients entering the study. The treatment allocation scheme is predefined and unchanged as patients enroll onto the study. These algorithms do not use any information on patients that have entered the trial. An example of a static algorithm is permuted blocks. In a permuted block randomization, blocks of random treatment assignments are generated within each stratification factor level *prior* to any patient enrolling on the trial. If there are two treatments, A and B, which are to be assigned equally, the number of A's is equal to the number of B's within a block. This algorithm is frequently used and should achieve a balance between the treatment arms within a stratification level. If however, there are several different stratification levels or several stratification factors, using permuted blocks can result in an overall treatment imbalance since all randomization is done within stratification factor level combination (Kernan, 1999; McEntegart, 2003).

An example of randomization using permuted blocks of size 4 can be found in Table 1. Suppose three patients have enrolled at the three stratification factor levels (9 total patients). Though treatment is balanced within stratification factor level, there is an overall treatment imbalance of 6 A's and 3 B's.

**Table 1:** Three sets of permuted blocks of size 4.

Stratification Factor Level	Permuted Block of Size 4 Treatment Allocation
1	AABBABABBAAB
2	AABBAABBAABB
3	ABABBABABBAA

Dynamic randomization algorithms aim to balance treatment arms with respect to stratification factor levels while keeping an overall treatment balance. Dynamic algorithms generate random treatment assignments based on the stratification levels of the patients that have entered the trial as well as the patient entering the study. For every patient entering a trial, a treatment assignment is generated by taking into account the stratification factor levels of each of the previously enrolled patients.

Minimization is a dynamic algorithm (Pocock & Simon, 1975). Based on stratification factor levels of patients currently on the trial and the treatments each of them is assigned, an imbalance score is computed for each available treatment. This imbalance score represents the imbalance that would be generated across treatments taking into account stratification factors levels *if* that treatment was assigned. In general, the treatment with the lowest imbalance score is then given preference when assigning treatments.

The SAS® program presented was written to test a new patient randomization system incorporating minimization being implemented within a nationwide oncology group. In order to perform this task, the new system randomly generates data for a specified number of test patients. The treatment assignment generated by the new system was tested using the SAS program.

**The Example:**

**The Example** used throughout the paper involves a trial with two treatment arms, A and B. The overall ratio of treatment A to treatment B will be 1:1, an equal allocation randomization. This example has two stratification factors used in the minimization. A patient is classified as one of two levels for the first stratification factor (Stratification Factor I), 5 or 6. Also, a patient is classified as one of two levels for the second stratification factor (Stratification Factor II), 3 or 4. The stratification factors are equally weighted in the calculation of the imbalance score. The test case is number 11017, the 17<sup>th</sup> patient to enroll in the study.

**Patient Data:**

Since minimization also depends on the enrollment of the previous patients, the treatment assignment and stratification levels for the first 16 patients enrolled on the study are included in Table 2. These patient data were randomly generated. The treatment assignments were assigned by the patient registration system. We will be using our SAS code to test the treatment assignment of the 17<sup>th</sup> patient, the TEST CASE.

**Table 2:** Stratification Factor Levels and Treatment Assignment of the first 17 patients

Treatment Assignment	Case	Stratification Factor I	Stratification Factor II
B	11001	6	4
A	11002	5	3
A	11003	5	3
B	11004	6	3
A	11005	5	4
B	11006	5	4
A	11007	6	4
B	11008	5	3
A	11009	6	4
A	11010	5	4
B	11011	6	3
A	11012	5	3
A	11013	5	3
B	11014	5	3
B	11015	5	4
B	11016	5	4
<b>B*</b>	<b>11017</b>	<b>5</b>	<b>3</b>

**\*TEST CASE:** This treatment assignment will be tested for accuracy. Treatment ‘B’ was assigned by the patient registration system. The objective is to test whether or not the treatment assignment is correct.

**Initial Calculations:**

For the calculations involved in minimization, the marginal sums by treatment by stratification level for the patients currently enrolled will be used. These sums are provided in Table 3.

**Table 3:** Breakdown of the first 16 patients enrolled on the study

Treatment	Stratification Factor I	Total	Stratification Factor II	Total
A	Five	6	Three	4
	Six	2	Four	4
B	Five	5	Three	4
	Six	3	Four	4

Since the TEST CASE (patient 17) has stratification factor I of 5 and stratification factor II of 3, only the breakdown of the patients enrolled prior to the TEST CASE (patient 17) with the same stratification factor levels will be needed for this calculation. The breakdown by stratification factor level is provided in Table 4.

**Table 4:** Breakdown of the first 16 patients with stratification factors the same as the TEST CASE

Treatment	Strat Factor I	Total	Strat Factor II	Total
A	Five	6	Three	4
B	Five	5	Three	4
<b>Total</b>		11		8

The next step is to determine the marginal totals if the TEST CASE is assigned to treatment A (Appendix A, line 130). Similar calculations are performed for Treatment B (Appendix A, line 170). Tables 5 and 6 give the breakdown for each potential treatment assignment.

**Table 5:** Breakdown if TEST CASE is assigned to treatment A (Potential Treatment A)

Treatment	Strat Factor I	Total	Stratification II	Total
A	Five	7	Three	5
B	Five	5	Three	4
<b>Total</b>		12		9

**Table 6:** Breakdown if TEST CASE is assigned to treatment B (Potential Treatment B)

Treatment	Strat Factor I	Total	Strat Factor II	Total
A	Five	6	Three	4
B	Five	6	Three	5
<b>Total</b>		12		9

The expected value for each treatment by stratification factor level is calculated. To do this, the proportion that should be randomized to a treatment arm is multiplied by the total number of patients within the stratification factor level. For **The Example**, the allocation to each treatment is 1:1. To calculate the proportion that should be randomized to treatment A, the quotient of the allocation to arm A and the sum of the allocation to all treatments is

calculated  $\left(\frac{1}{1+1} = \frac{1}{2}\right)$  (Appendix A, line 270). Thus the expectation for stratification factor I is  $6\left(12 \times \frac{1}{2}\right)$

(Appendix A, line 280). Similarly for stratification factor II, the expected is  $4.5 \left( 9 \times \frac{1}{2} \right)$ . The difference between the expected and the observed values is then taken (Appendix A, line 300). These calculations are illustrated in Tables 7 and 8.

**Table 7: Breakdown if TEST CASE was assigned to treatment A with expectations and differences (Potential Treatment A)**

Treatment	Strat Factor I	Expectation # pts * ratio	Difference (Observed-Expected)	Strat Factor II	Expectation n	Difference (Observed-Expected)
<b>A</b>	<b>Five</b>	$\left( 12 \times \frac{1}{2} \right) = 6$	<b>7-6=1</b>	<b>Three</b>	$\left( 9 \times \frac{1}{2} \right) = 4.5$	<b>5-4.5=0.5</b>
<b>B</b>	<b>Five</b>	$\left( 12 \times \frac{1}{2} \right) = 6$	<b>5-6=-1</b>	<b>Three</b>	$\left( 9 \times \frac{1}{2} \right) = 4.5$	<b>4-4.5=-0.5</b>

**Table 8: Breakdown if TEST CASE was assigned to treatment B with expectations and differences (Potential Treatment B)**

Treatment	Strat Factor I	Expectation # pts * ratio	Difference (Observed-Expected)	Strat Factor II	Expectation n	Difference (Observed-Expected)
<b>A</b>	<b>Five</b>	$\left( 12 \times \frac{1}{2} \right) = 6$	<b>6-6=0</b>	<b>Three</b>	$\left( 9 \times \frac{1}{2} \right) = 4.5$	<b>4-4.5=-0.5</b>
<b>B</b>	<b>Five</b>	$\left( 12 \times \frac{1}{2} \right) = 6$	<b>6-6=0</b>	<b>Three</b>	$\left( 9 \times \frac{1}{2} \right) = 4.5$	<b>5-4.5=0.5</b>

**Distance Measures and Imbalance Scores:**

An intermediate step in minimization is to calculate an imbalance score using the differences for each potential treatment. The imbalance score is a summary measure of the lack of balance created by the potential treatment assignment. The distance measure is a way of measuring how far the potential treatment allocation is from the 'ideal' allocation. Distance, however, can be measured in several different ways. This program uses three such measures: RANGE, VARIANCE and MAX.

The RANGE distance measure is defined to be the absolute value of the difference between the differences for each stratification factor (Appendix A, line 320). The VARIANCE distance measure is defined to be the variance of the differences (Appendix A, line 330). Similarly, the MAX distance measure is defined to be the maximum of the differences (Appendix A, line 340) (Gentleman, 2001).

For example in Table 8, for stratification I, the (observed – expected) for each treatment is 0, thus the RANGE of these values is the absolute value of the difference between 0 and 0 which is 0. For stratification II, the (observed – expected) for treatment A is 0.5 and for B is -0.5, thus the RANGE of these values is the absolute value of the difference between 0.5 and (-0.5) which is 1. See Table 9 for distance calculations for each distance measure.

In **The Example**, RANGE was the distance measure used. Therefore the result of distance measure calculations for potential treatment A are 2 and 1 for stratification factors I and II respectively; and for potential treatment B are 0 and 1 for stratification factors I and II respectively.

**Table 9:** Distance for each stratification factor for each Potential Treatment Assignment

Potential Treatment	Distance Measure	Strat Factor	
		I	II
A	RANGE	$ABS[1 - (-1)] = 2$	$ABS[0.5 - (-0.5)] = 1$
B		$ABS[0-0]=0$	$ABS[0.5 - (-0.5)] = 1$
A	VARIANCE	$[1^2 + (-1)^2]/2=1$	$[0.5^2 + (-0.5)^2]/2=0.25$
B		$[0^2 + 0^2]/2=0$	$[0.5^2 + (-0.5)^2]/2=0.25$
A	MAXIMUM	$MAX(1,-1)=1$	$MAX[0.5, -0.5]=0.5$
B		$MAX[0,0]=0$	$MAX[0.5,(-0.5)]=0.5$

Once the distance for each stratification factor has been determined, the distances from each stratification factor need to be combined to create an imbalance score (Appendix A, line 350). Since stratification factors are weighted equally in **The Example**, the distances for each stratification factor are added together to create an overall imbalance score for each potential treatment. The calculations of imbalance scores for all distance measures are summarized in Table 10.

In **The Example**, the imbalance score for potential treatment A is 3 and the imbalance score for potential treatment B is 1.

**Table 10:** Distance Calculations and Imbalance scores

Distance Measure				Range		Variance		Maximum	
Potential Treatment Assignment		Treatment	Observed-Expected	Distance	Imbalance Score	Distance	Imbalance Score	Distance	Imbalance Score
				A	SF I	A	1	range (1,-1) =2	Ga=3.0
B	-1								
SF II	A	0.5	range (0.5,-0.5) =1		var (0.5,-0.5) =0.25	max (0.5,-0.5) =0.5			
	B	-0.5							
B	SF I	A	0	range (0,0) =0	Gb=1.0	var (0,0) =0	Gb=0.25	max (0,0) =0	Gb=0.5
		B	0						
	SF II	A	-0.5	range (-0.5,0.5) =1		var (-0.5,0.5) =0.25		max (-0.5,0.5) =0.5	
		B	0.5						

**Treatment Selection Method:**

Once a minimization imbalance score has been calculated for each potential treatment, a treatment must be chosen based on the scores. The program contains three methods for selecting treatment. The first is called BEST in the SAS program (Appendix A, line 360). This method always assigns the treatment with the smallest imbalance score to the enrolling patient. This is the treatment that will generate the least amount of imbalance in treatment arms based on the stratification factor levels of that patient. In **The Example**, BEST is used to select treatment. The imbalance score for treatment A is 3 and the imbalance score for treatment B is 1. Since  $1 < 3$ , treatment B is assigned which matches the patient registration system assignment.

The second method is called PROB (Appendix A, line 380). This method ranks the imbalance scores and then assigns each treatment with a specified probability according to rank. The treatment with the lowest score is assigned with a specified probability, the treatment with the second to lowest score with a specified probability, and so on. If there are three treatments, at the beginning of the study three probabilities will be specified. If there are two treatments, at the beginning of the study two probabilities will be specified.

For example, there are two treatments in **The Example**. If PROB had been used to select treatment, two probabilities would have been specified, such as 0.75 and 0.25. Since the treatment with the lowest imbalance score is B (with an imbalance score of 1), treatment B would be assigned with a probability of 0.75. Treatment A has the next lowest imbalance score and would therefore be assigned with probability 0.25. Typically, the probability assigned to selecting the treatment with the lowest score is higher than the probability assigned to selecting the treatment with the next lowest scores, etc. The treatment that, if assigned, would generate the smallest amount of imbalance would be selected with the highest probability and a treatment that, if assigned, would generate the greatest amount of imbalance would be selected with a lower probability. Assigning probability 1 to the treatment with the smallest imbalance score and probability 0 to all other treatments is equivalent to the BEST method.

The third method is called PROP (Appendix A, line 370). This method is similar to PROB, in that treatments are selected with varying probabilities. However, PROP assigns probabilities based on the value of the imbalance score. The treatments are assigned with a probability that is inversely proportional to the level of imbalance that the treatment would generate. In **The Example**, the imbalance score for treatment A is 3 and the imbalance score for treatment B is 1. Therefore, if PROP had been used to select the treatment, treatment A would be assigned with probability  $1 - \frac{3}{1+3} = 0.25$  and treatment B would be assigned with probability  $1 - \frac{1}{1+3} = 0.75$ .

Table 11 illustrates how treatments are selected using the three methods.

**Table 11:** Examples of Imbalance Scores and Treatment Selections using BEST, PROB, and PROP

Potential Treatment	Imbalance Score	Action for each Treatment Selection Method		
		BEST	PROB (0.75, 0.25)	PROP
A	3.0	Assign B	Assign A with p= 0.25 Assign B with p= 0.75	Assign A with p=0.25 Assign B with p= 0.75
B	1.0			
A	1.25	Assign B	Assign A with p= 0.25 Assign B with p= 0.75	Assign A with p= 0.17 Assign B with p=0.83
B	0.25			
A	1.5	Assign B	Assign A with p= 0.25 Assign B with p= 0.75	Assign A with p= 0.25 Assign B with p=0.75
B	0.5			

### Random Component in Treatment Selection:

In both the PROB and the PROP treatment selection methods, a treatment is randomly assigned with certain probability. The random aspect of this assignment is handled through the patient registration system. The system generates a random number between 0 and 1 for the enrolling patient. The SAS code pulls the random number from the database and it is compared to the specified probabilities used to assign treatments (Appendix A, line 503 for PROB; line 378 for PROP). Suppose the probability of assigning treatment A is 0.25 and the probability of assigning treatment B is 0.75. If the random number generated by the system is between 0 and 0.25, treatment A will be assigned. If the random number is greater than 0.25 and is less than 1, treatment B is assigned.

### Data Necessary to Run the SAS Code:

Information about the protocol and randomization scheme is necessary to perform minimization and run the SAS code. The information about the randomization scheme that is needed is: the number of treatments, the treatment allocation ratios, the number of stratification factors, the weights assigned to each stratification factor, the distance measure, and the treatment selection method. In addition, in order to run the SAS code, we need additional information, specifically: the number of patients for which data has been randomly generated, the test case, the random number generated by the patient registration system when a probability process is used to determine treatment selection (PROB, PROP), and the probabilities for each imbalance score (used only when using the PROB treatment selection method).

**Other Options:**

Although **The Example** illustrates a simple case of minimization, the SAS code can accommodate several other situations. Some possibilities are: unequal treatment allocation, unequal weights for stratification factors, greater than two treatments, greater than two stratification factors, greater than two levels within a stratification factor.

*Unequal Weights:*

The weights option in the SAS code (Appendix A, line 30) pertains to the weight given to each stratification factor when calculating the imbalance score. In **The Example**, each stratification factor was assigned equal weight, however, unequal weights could also have been used. For example, suppose the weight for stratification factor I was 2 and for stratification factor II was 3 and the RANGE for stratification factor I was 2 and the RANGE for stratification factor II was 1, the imbalance score would be 7 ( $2 \times 2 + 3 \times 1$ ). The weight option can be used if one stratification factor is deemed of greater importance than another.

*Unequal Treatment Allocation:*

Unequal treatment allocation (Appendix A, line 10) can be specified if the protocol requires more patients to be randomized to one or more treatments. An example of unequal allocation would be a 2:1 randomization to A and B, respectively. This would randomize twice as many patients to treatment A as to treatment B.

*Number of treatments:*

In **The Example**, there are only 2 treatments that can be assigned. Code has been generated for three treatments or more (expanded code not shown).

*Number of Stratification Factors:*

In **The Example**, there are 2 stratification factors that are used for our calculations. Similar calculations for each stratification factor would be done if there were more than 2 stratification factors (expanded code not shown).

*Number of Stratification Levels within a factor:*

In **The Example**, each stratification factor had two levels. The code provided can accommodate any number of stratification factor levels. The calculations are independent of the number of stratification factor levels.

**Conclusions:**

This SAS code can be used to test randomization allocations made using minimization. The SAS code along with **The Example** also illustrate how the calculations for minimization are carried out. In several situations where randomization is used there are factors known to influence treatment effect. For this reason minimization may be a reasonable method to balance the factors across treatment arms.

Randomization is a necessary component for clinical trials with more than one treatment. Though the use of dynamic randomization algorithms still seem to be under debate (CPMP, 2004; Weir, 2003), there are undeniable benefits. As always, the method used for randomization needs to be considered during analysis of results.

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 registered trademarks or trademarks of their respective companies.

#### References:

Committee for Proprietary Medicinal Products (CPMP) (2004). Points to Consider on Adjustment for Baseline Covariates. *Statistics in Medicine*, Volume 23, pp. 701-709.

Everitt, Brian (1994). *Statistical Methods for Medical Investigations*. John Wiley and Sons, Inc. New York.

Gentleman, Robert (2001). PRS: A Patient Randomization System. DFCI Biostatistical Science Department Internal Document.

Kernan WN, Viscoli CM, Makuch RW, Brass LM, Horwitz RI (1999). Stratified Randomization for Clinical Trials. *Journal of Clinical Epidemiology*, Volume 52 (1), pp. 19-26.

McEntegart, Damian J (2003). The Pursuit of Balance Using Stratified and Dynamic Randomization Techniques: An Overview. *Drug Information Journal*, Volume 37, pp. 293-308.

Pocock SJ, Simon R (1975). Sequential Treatment Assignment with Balancing for Prognostic Factors in the Controlled Clinical Trial. *Biometrics* Vol. 31, pp. 103-115.

Simon R, Wittes RE, Ellenburg SS (1985). Randomized Phase II Clinical Trials. *Cancer Treatment Reports*, Vol. 69, No 12 (December), pp. 1375-1381.

Weir CJ, Kennedy RL (2003). Comparison of Stratification and Adaptive Methods for Treatment Allocation in an Acute Stroke Clinical Trial. *Statistics in Medicine*, Volume 22, pp. 705-726.

Thanks to Edie Weller, Robert Gentleman and Bob Gray (DFCI) for their statistical expertise and their support through the testing of this project. Thanks to AJ Bozogian and Carol Chami (Frontier Science) for their programming expertise as well as knowledge of the Patient Registration System. Thanks also to the Boston SAS Users Group for providing their many ideas, SAS techniques and expertise.

**Contact Information:** Donna Levy & Emily Blood  
Dana-Farber Cancer Institute  
44 Binney Street, Biostatistical Science Department  
Boston MA 02215  
[levy@jimmy.harvard.edu](mailto:levy@jimmy.harvard.edu), [blood@jimmy.harvard.edu](mailto:blood@jimmy.harvard.edu)

## Appendix A:

```

*****;
* NESUG Submission: 2004      *;
* Emily Blood and Donna Levy *;
* May 6, 2004                *;
* TWO treatments            *;
*****;

*****;
* The SAS code was used to test a new patient randomization system at our institute. ;
* The new system allows for several types of randomization methods. However, this ;
* portion of the code was used to test features related to minimization. ;
* ;
* Minimization is a dynamic randomization algorithm. It is considered dynamic as it ;
* depends on the current patient being entered onto the study thus the treatment scheme ;
* cannot be laid out beforehand. Minimization also depends on the patient ;
* characteristics and treatment assignments of previous patients. ;
* ;
* The goal of minimization is to have the least imbalance between treatments taking ;
* stratification factors into account. ;
* ;
* SEE README FILE FOR FULL DESCRIPTION OF GLOBAL MACRO VARIABLES AND THEIR FUNCTION ;
* ;
* Program: ;
*   minimizaton.sas ;
* Other necessary files (Indirectly): ;
*   p2400.pro: eligibility criteria file for protocol ;
*   p2400.xml: patient randomization file for protocol ;
*   README: descriptor of macro variables used in program ;
*****;

*****;
%LET directory= F:\Presentations\NESUG\2004\2Treatment;
*****;

OPTIONS NOCENTER LS=80 SGEN MERROR;
LIBNAME SASData '.';
*****;
*                               TO BE PULLED FROM XML AND/OR MINIMIZATION TABLE ;
*                               OR ENTERED MANUALLY ;
*****;
***** step of XML where MINIMIZATION is being used and tested;
%LET step=1;
%LET step=2;

***** protocol that is being tested;
%LET prot=P2400;

***** number of patients for which data is to be generated;
%LET numberpt=17;

***** treatment allocation;
***** if 1:1 or 2:2 there will be equal allocation to the possible treatments;
%LET treatallocA=1; * line 10;
%LET treatallocB=1;

***** number of treatments;
%LET numtreat=2; * line 20;

***** sequence numbers assigned to enrolling pts;
* start pulling data from specified sequence number;
%LET seqnost=11001;

***** sequence numbers to end pulling data (inclusive);
%LET seqnoend=110&numberpt;

***** strat factor weights used in Gi;
***** XML: MINIMIZER FACTFUN WEIGHT=1, WEIGHT=1;
%LET weight1=1; * line 30;
%LET weight2=1;

```

```

**** number of strat factors in prot;
**** XML: NUMFACTS;
%LET numstrat=2; * line 40;

**** treatment selection method from minchoose statement;
**** XML: MINCHOOSE METHOD;
%LET selectionmethod='BEST'; * line 50;
* %LET selectionmethod='PROP';
* %LET selectionmethod='PROBS';

**** from minmethod statement;
**** XML: MINMETHOD METHOD= WILL HAVE A METHOD FOR EACH STRAT FACTOR;
**** XML: FOR THIS TESTING, THEY WILL ALWAYS BE THE SAME METHOD FOR ALL STRAT FACTORS; **** IN ONE
FILE;
%LET method='range'; * line 60;
* %LET method='variance';
* %LET method='max';

**** from .min file;
**** the random number (f) that was generated for each case (f## where ## is case;
**** number);
**** to determine treatment that will be assigned;
%LET f17=0.044297; * line 70;
%LET f18=0.523362;
%LET f73=0.887665;

**** Probabilities used to calculate the props for each treatment;
%LET prob1=0.4; * line 80;
%LET prob2=0.6;

***** CASE 1 THROUGH n-1 FAKE DATA *****;
* PATIENT TABLES This data will be randomly generated when we start testing;
* However seeds to generate data, will be maintained so that tests can be ;
* repeated. ;
* The data is pulled from prsheader & prspatdat. ;
* A permanent SAS data set has been created. ;

DATA Generate; * line 90;
  SET SASData.&prot.data;
  WHERE step=&step AND (protocol="&prot") AND (seqno GE &seqnost AND seqno LE &seqnoend);
  case=seqno;

  %MACRO StratFactors;
    %DO j= 1 %TO &numstrat;
      FORMAT fac&j stratfct&j 1.;
      stratfct&j=fac&j/1;
    %END;
  %MEND StratFactors;
  %StratFactors;

KEEP case treatment stratfct1 stratfct2;
RUN;

***** DATA SET: TEST CASE creating macro variables *****;
DATA CaseN;
  SET Generate;
  IF case=&seqnoend;
  CALL SYMPUT ('caseNstrat1', TRIM(LEFT(PUT(stratfct1, 1.)))); * line 100;
  CALL SYMPUT ('caseNstrat2', TRIM(LEFT(PUT(stratfct2, 1.))));
  CALL SYMPUT ('caseNprstreat', TRIM(LEFT(PUT(treatment, 1.))));
RUN;
***** DATA SET: CASE 1 THROUGH N *****;
DATA IncludeCaseN; * line 110;
  SET Generate;
  prsassignedtreat=treatment;
  IF case=&seqnoend THEN case=99;
RUN;
*****DATA SET: ONLY CASE N reassigned variable names *****;
DATA OnlyCaseN; * line 120;
  SET IncludeCaseN;
  IF case=99;
  obs=_N_;
  KEEP case prsassignedtreat;
RUN;

```

```

***** IF Case=N ASSIGNED TREATMENT A *****;
DATA AssignA;
  SET IncludeCaseN;
  IF case=99 THEN treatment='A';
  RUN;
***** Strat 1 *****;
* counts for strat factor 1, if case N assigned treatment A;
PROC FREQ DATA=AssignA NOPRINT;
  TABLES treatment* stratfct1 /OUT=NumAssignA_SF1;
  WHERE stratfct1=&caseNstrat1;
RUN;

DATA NumAssignA_SF1;
  SET NumAssignA_SF1;
  N=_N_;
  case=99;
RUN;

%flatten(indata=NumAssignA_SF1, outdata=FlatNumAssignA_SF1, byvar=N, casevar=case );

DATA FlatNumAssignA_SF1;
  SET FlatNumAssignA_SF1;
  IF count_1=. THEN count_1=0;
  IF count_2=. THEN count_2=0;
  num_A_A_sf1_&caseNstrat1=count_1;
  num_B_A_sf1_&caseNstrat1=count_2;
  KEEP case num_A_A_sf1_&caseNstrat1 num_B_A_sf1_&caseNstrat1;
RUN;
***** Strat 2 *****;
* counts for strat factor 2, if case N assigned treatment A;
PROC FREQ DATA=AssignA NOPRINT;
  TABLES treatment* stratfct2 /OUT=NumAssignA_SF2;
  WHERE stratfct2=&caseNstrat2;
RUN;

DATA NumAssignA_SF2;
  SET NumAssignA_SF2;
  N=_N_;
  case=99;
RUN;

%flatten(indata=NumAssignA_SF2, outdata=FlatNumAssignA_SF2, byvar=N, casevar=case );

DATA FlatNumAssignA_SF2;
  SET FlatNumAssignA_SF2;
  IF count_1=. THEN count_1=0;
  IF count_2=. THEN count_2=0;
  num_A_A_sf2_&caseNstrat2=count_1;
  num_B_A_sf2_&caseNstrat2=count_2;
  KEEP case num_A_A_sf2_&caseNstrat2 num_B_A_sf2_&caseNstrat2;
RUN;
* totals for strat 1 and strat 2 if case N assigned to treatment A;
* to be used in Ga calculations;
DATA TotalAassign;
  MERGE FlatNumAssignA_SF1 FlatNumAssignA_SF2;
  BY case;
RUN;
***** IF Case=N ASSIGNED TREATMENT B *****;
DATA AssignB;
  SET IncludeCaseN;
  IF case=99 THEN treatment='B';
  RUN;
***** Strat 1 *****;
* counts for strat factor 1, if case N assigned treatment B;
PROC FREQ DATA=AssignB NOPRINT;
  TABLES treatment* stratfct1 /OUT=NumAssignB_SF1;
  WHERE stratfct1=&caseNstrat1;
RUN;

DATA NumAssignB_SF1;
  SET NumAssignB_SF1;
  N=_N_;
  case=99;
RUN;

```

```

%flatten(indata=NumAssignB_SF1, outdata=FlatNumAssignB_SF1, byvar=N, casevar=case );
DATA FlatNumAssignB_SF1; SET FlatNumAssignB_SF1;
IF count_1=. THEN count_1=0;
IF count_2=. THEN count_2=0;
  num_A_B_sf1_&caseNstrat1=count_1;
  num_B_B_sf1_&caseNstrat1=count_2;
KEEP case num_A_B_sf1_&caseNstrat1 num_B_B_sf1_&caseNstrat1;
RUN;

***** Strat 2 *****;
* counts for strat factor 2, if case N assigned treatment B;
PROC FREQ DATA=AssignB NOPRINT; * line 200;
  TABLES treatment* stratfct2 /OUT=NumAssignB_SF2;
  WHERE stratfct2=&caseNstrat2;
RUN;

DATA NumAssignB_SF2;
  SET NumAssignB_SF2;
  N=_N_;
  case=99;
RUN;

%flatten(indata=NumAssignB_SF2, outdata=FlatNumAssignB_SF2, byvar=N, casevar=case );

DATA FlatNumAssignB_SF2;
  SET FlatNumAssignB_SF2;
IF count_1=. THEN count_1=0;
IF count_2=. THEN count_2=0;
  num_A_B_sf2_&caseNstrat2=count_1;
  num_B_B_sf2_&caseNstrat2=count_2;
KEEP case num_A_B_sf2_&caseNstrat2 num_B_B_sf2_&caseNstrat2;
RUN;

/* num_A_B_sf1_2=      The number of patients who have
                       strat factor1=2
                       who are on treatment A
                       when case N is assigned to treatment B

   num_A_B_sf1_1=      The number of patients who have
                       strat factor1=1
                       who are on treatment A
                       when case N is assigned to treatment B

do for all strat and treatment combos
*/
* totals for strat 1 and strat 2 if case N assigned to treatment B;
* to be used in Gb calculations;

DATA TotalBassign; * line 250;
  MERGE FlatNumAssignB_SF1 FlatNumAssignB_SF2;
  BY case;
RUN;

DATA Together; * line 260;
  MERGE TotalAassign TotalBassign OnlyCaseN;
  BY case;
method=&method;

* expected could also be calculated using (num_A_B_sf1_&caseNstrat1 + num_B_B_sf1_&caseNstrat1);
* both calculations produce the same value;
ratioA= &treatallocA/(&treatallocA + &treatallocB); * line 270;
ratioB= &treatallocB/(&treatallocA + &treatallocB);

* strat factor 1;
expectedsf1= (num_A_A_sf1_&caseNstrat1 + num_B_A_sf1_&caseNstrat1); * line 280;
expectedsf1_A= (expectedsf1)* ratioA;
expectedsf1_B= (expectedsf1)* ratioB;

* strat factor 2;
expectedsf2= (num_A_A_sf2_&caseNstrat2 + num_B_A_sf2_&caseNstrat2); * line 290;
expectedsf2_A= (expectedsf2)* ratioA;
expectedsf2_B= (expectedsf2)* ratioB;

```

```

* if A is assigned;
  z_A_A_sf1_&caseNstrat1= (num_A_A_sf1_&caseNstrat1 - expectedsf1_A);      * line 300;
  z_B_A_sf1_&caseNstrat1= (num_B_A_sf1_&caseNstrat1 - expectedsf1_B);
  z_A_A_sf2_&caseNstrat2= (num_A_A_sf2_&caseNstrat2 - expectedsf2_A);
  z_B_A_sf2_&caseNstrat2= (num_B_A_sf2_&caseNstrat2 - expectedsf2_B);

* if B is assigned;
  z_A_B_sf1_&caseNstrat1= (num_A_B_sf1_&caseNstrat1 - expectedsf1_A);      * line 310;
  z_B_B_sf1_&caseNstrat1= (num_B_B_sf1_&caseNstrat1 - expectedsf1_B);
  z_A_B_sf2_&caseNstrat2= (num_A_B_sf2_&caseNstrat2 - expectedsf2_A);
  z_B_B_sf2_&caseNstrat2= (num_B_B_sf2_&caseNstrat2 - expectedsf2_B);

*****;
*
                      METHOD = RANGE
*****;
IF method='range' THEN DO;                                          * line 320;
  * difference if A is assigned;
  maxA_sf1_&caseNstrat1=MAX(z_A_A_sf1_&caseNstrat1, z_B_A_sf1_&caseNstrat1);
  minA_sf1_&caseNstrat1=MIN(z_A_A_sf1_&caseNstrat1, z_B_A_sf1_&caseNstrat1);
  maxA_sf2_&caseNstrat2=MAX(z_A_A_sf2_&caseNstrat2, z_B_A_sf2_&caseNstrat2);
  minA_sf2_&caseNstrat2=MIN(z_A_A_sf2_&caseNstrat2, z_B_A_sf2_&caseNstrat2);
  d_A_sf1_&caseNstrat1= ABS(maxA_sf1_&caseNstrat1 - minA_sf1_&caseNstrat1);
  d_A_sf2_&caseNstrat2= ABS(maxA_sf2_&caseNstrat2 - minA_sf2_&caseNstrat2);

  * difference if B is assigned;
  maxB_sf1_&caseNstrat1=MAX(z_A_B_sf1_&caseNstrat1, z_B_B_sf1_&caseNstrat1);
  minB_sf1_&caseNstrat1=MIN(z_A_B_sf1_&caseNstrat1, z_B_B_sf1_&caseNstrat1);
  maxB_sf2_&caseNstrat2=MAX(z_A_B_sf2_&caseNstrat2, z_B_B_sf2_&caseNstrat2);
  minB_sf2_&caseNstrat2=MIN(z_A_B_sf2_&caseNstrat2, z_B_B_sf2_&caseNstrat2);
  d_B_sf1_&caseNstrat1= ABS(maxB_sf1_&caseNstrat1 - minB_sf1_&caseNstrat1);
  d_B_sf2_&caseNstrat2= ABS(maxB_sf2_&caseNstrat2 - minB_sf2_&caseNstrat2);
END;

*****;
*
                      END METHOD = RANGE
*****;

*****;
*
                      METHOD = MAX
*****;

IF method='max' THEN DO;                                          * line 330;
  * difference if A is assigned;
  d_A_sf1_&caseNstrat1= MAX(z_A_A_sf1_&caseNstrat1, z_B_A_sf1_&caseNstrat1);
  d_A_sf2_&caseNstrat2= MAX(z_A_A_sf2_&caseNstrat2, z_B_A_sf2_&caseNstrat2);
  * difference if B is assigned;
  d_B_sf1_&caseNstrat1= MAX(z_A_B_sf1_&caseNstrat1, z_B_B_sf1_&caseNstrat1);
  d_B_sf2_&caseNstrat2= MAX(z_A_B_sf2_&caseNstrat2, z_B_B_sf2_&caseNstrat2);
END;

*****;
*
                      END METHOD = MAX
*****;

*****;
*
                      METHOD = VARIANCE
*****;

IF method='variance' THEN DO;                                     * line 340;
  * d (variance) if A is assigned;
  d_A_sf1_&caseNstrat1= ( (z_A_A_sf1_&caseNstrat1 )**2
                        + (z_B_A_sf1_&caseNstrat1 )**2 )/(&numtreat);
  d_A_sf2_&caseNstrat2= ( (z_A_A_sf2_&caseNstrat2 )**2
                        + (z_B_A_sf2_&caseNstrat2 )**2 )/(&numtreat);

  * d (variance) if B is assigned;
  d_B_sf1_&caseNstrat1= ( (z_A_B_sf1_&caseNstrat1 )**2
                        + (z_B_B_sf1_&caseNstrat1 )**2 )/(&numtreat);
  d_B_sf2_&caseNstrat2= ( (z_A_B_sf2_&caseNstrat2 )**2
                        + (z_B_B_sf2_&caseNstrat2 )**2 )/(&numtreat);

END;

*****;
*
                      END METHOD = VARIANCE
*****;

```

```

Ga= (&weight1) * d_A_sf1_&caseNstrat1 + (&weight2) * d_A_sf2_&caseNstrat2; * line 350;
Gb= (&weight1) * d_B_sf1_&caseNstrat1 + (&weight2) * d_B_sf2_&caseNstrat2;
selectionmethod=&selectionmethod;

*****;
*                               MINMETHOD = BEST                               ;
*****;
IF &selectionmethod='BEST' THEN DO; * line 360;
  FORMAT treatmentselection $3.;
  IF ga<gb THEN treatmentselection='A';
  ELSE IF ga >gb THEN treatmentselection='B';
  ELSE IF ga=gb THEN treatmentselection='tie';
END;
*****;
*                               MINMETHOD = PROP                               ;
*****;
IF &selectionmethod='PROP' THEN DO; * line 370;
  FORMAT treatmentselection $3.;

  IF ga=0 THEN ga=ga + 0.01;
  IF gb=0 THEN gb=gb + 0.01;
  propga= gb/(ga+gb);
  propgb= ga/(ga+gb);

  FORMAT random 8.6;
  random= &&f&numberpt;

  IF random LE propga THEN treatmentselection='A';
  ELSE IF random > propga THEN treatmentselection='B';
END;
RUN;

*****;
*                               MINMETHOD = PROBS                               ;
*****;
DATA Scores; * line 380;
  SET Together;
  score=Ga; Treatment='A'; OUTPUT;
  score=Gb; Treatment='B'; OUTPUT;
RUN;

DATA Probs; * line 390;
  prob=&prob1; OUTPUT;
  prob=&prob2; OUTPUT;
RUN;

PROC SORT DATA=Scores; BY score; RUN;
PROC SORT DATA=Probs; BY DESCENDING prob; RUN;

DATA Scores; SET Scores; obs=_N_; RUN;
DATA Probs; SET Probs; obs=_N_; RUN;
PROC SORT DATA=Scores; BY obs; RUN;
PROC SORT DATA=Probs; BY obs; RUN;

DATA ScoreProbs; MERGE Scores Probs; BY obs; case=1;
KEEP obs case treatment prob score; RUN;

%FLATTEN(INDATA=ScoreProbs, OUTDATA=ScoreProbsFlat, BYVAR=obs, CASEVAR=case);

DATA ScoreProbsFlat; * line 500;
  SET ScoreProbsFlat;
  FORMAT random 8.6;
  random= &&f&numberpt;

  sum1=prob_1;
  sum2=sum1 + prob_2;

  IF random LE sum1 THEN treatmentselection=treatment_1;
  ELSE IF random GT sum1 THEN treatmentselection=treatment_2;
RUN;

```

```

*****;
*                               OUTPUT                               *;
*****;
TITLE1"&prot &directory";                                     * line 600;
TITLE2 "First &numberpt Cases";
PROC PRINT DATA=Generate N NOOBS;
RUN;

PROC FREQ DATA=Generate;
TABLES stratfct1*stratfct2;
RUN;

TITLE2"Counts by Treatment by strat factor, excluding Test Case= &seqnoend";
PROC FREQ DATA=Generate;
TABLES treatment * (stratfct1 stratfct2) ;
WHERE case NE &seqnoend;
RUN;

TITLE2 "Test Case Only (&seqnoend)";
PROC PRINT DATA=CaseN N NOOBS ; RUN;

TITLE2 "Data Including Test Case (&seqnoend)";
PROC PRINT DATA=IncludeCaseN N NOOBS ;
RUN;

*****;
*                               IF A ASSIGNED                               *;
*****;
TITLE2'Counts by Treatment for strat factor 1, if case N assigned treatment A';
PROC FREQ DATA=AssignA ;
  TABLES treatment* stratfct1 /OUT=NumAssignA_SF1;
  WHERE stratfct1=&caseNstrat1;
RUN;

TITLE2'Counts by Treatment for strat factor 2, if case N assigned treatment A';
PROC FREQ DATA=AssignA;
  TABLES treatment* stratfct2 /OUT=NumAssignA_SF2;
  WHERE stratfct2=&caseNstrat2;
RUN;

TITLE2'DATA=Total A assign';
PROC PRINT DATA=TotalAassign N NOOBS;
RUN;

*****;
*                               IF B ASSIGNED                               *;
*****;
TITLE2'Counts by Treatment for strat factor 1, if case N assigned treatment B';
PROC FREQ DATA=AssignB;
  TABLES treatment* stratfct1 /OUT=NumAssignB_SF1;
  WHERE stratfct1=&caseNstrat1;
RUN;

TITLE2'Counts by Treatment for strat factor 2, if case N assigned treatment B';
PROC FREQ DATA=AssignB;
  TABLES treatment* stratfct2 /OUT=NumAssignB_SF2;
  WHERE stratfct2=&caseNstrat2;
RUN;

TITLE2'DATA=Total B assign';
PROC PRINT DATA=TotalBassign N NOOBS;
RUN;

*****;
*                               IMBALANCE SCORES AND TREATMENT ASSIGNMENT                               *;
*****;
OPTIONS LS=256; PROC TRANSPOSE DATA=Together OUT=TogetherT; BY Case; RUN;

TITLE2 'Calculation Data';
PROC PRINT DATA=TogetherT NOOBS LABEL ;
  LABEL _name_='Minimization Variable' coll='Value';
  VAR _name_ coll;
RUN;

PROC PRINT DATA=ScoreProbsFlat N NOOBS; RUN;

```