Randomizing Small strata Groups - Balancing Baseline Population Measures While Staying True to the Selected Allocation Strategy An Alternative to PROC SURVEYSELECT

Gregg M. Oginz, Health Dialog, Boston, MA

ABSTRACT

Essential components of a Stratified Randomized Control Trial are that, with respect to pre-period outcome measures, strata means are significantly different, study arm means are not statistically different and study arm populations stay true to the selected allocation strategy. When employed correctly, a sound stratification methodology will establish well balanced baseline populations.

In simple terms, PROC SURVEYSELECT's method of randomizing stratum is to count out in repetition, arm assignments in accordance with the selected allocation. To illustrate, for a 4:1 allocation, 1,1,1,1,0 would be counted out and repeated over a 'shuffled' stratum. When a stratum consists of five members, four are assigned to "Study Arm 1" and one to "Study Arm 2". But what happens when a stratum consists of nine members? PROC SURVEYSELECT counts out 1,1,1,1,0,1,1,1,1 over the nine 'shuffled' members. The outcome is eight members assigned to "Study Arm 1" and one to "Study Arm 2". The consequence is deviation from the selected allocation and the potential for unbalanced baseline measures over the two arms. This paper explores an alternative approach.

INTRODUCTION

The benefit of stratified random sampling is that clusters of similar records (or stratum) get randomized as sub populations, allowing the researcher to randomize an entire study population at more granular level. This stratified approach helps ensure that after all records have been randomized; baseline study arm populations, with respect to the primary outcome measure(s), are as similar to each other as possible. This empowers the analyst to detect a true intervention effect when present.

Critical in stratified random sampling, is staying true to the selected allocation strategy. For example, if a 4:1 allocation is determined best for a two arm study design, then the resulting study arm populations should contain 80% and 20% of the membership respectively. It is here that PROC SURVEYSELECT has a potential of falling short.

PROC SURVEYSELECT (SINGLE RANDOMIZATION EXAMPLE)

A dataset "to_be_randomized" was prepared with twenty-three members (a-w) over three strata groupings (1,2,3) and was run through the procedure code shown below. The objective of this randomization is to develop two similar study arms with respect to the primary outcome measure 'Score'. The output dataset "randomized" is presented by **Table 1** on the following page.

```
proc surveyselect data=to_be_randomized
 samprate = .80
 out = randomized
 method = SRS
 outall;
 strata strata_group; run;
```

To simply illustrate what happens behind the scenes, PROC SURVEYSELECT parses out each strata group one at a time, shuffles the members within the stratum, counts out arm assignments in a sequential pattern in-line with the allocation strategy and finally returns the members in their original sort order. For example a 4:1 allocation would be counted out as 1,1,1,1,0 and repeated over the 'shuffled' stratum. In other words for every multiple of five members, what I refer to as a "Complete Allocation Unit" (CAU), we stay true to the allocation strategy. That is four of the population members will be assigned to "Study Arm 1" and one to "Study Arm 2", a true 80/20 split.

On **Table 1**, we see there are five members where 'strata_group' is equal to "1", a complete allocation unit. Applying the logic previously described we would expect four of these members to be assigned to "Study Arm 1" and one member to "Study Arm 2". As predicted this is the case, four members have 'Selected' equal to "1" (Study Arm 1) and one member has 'Selected' equal to "0" (Study Arm 2). PROC SURVEYSELECT has stayed true to the statement samprate = .80 (a 4:1 allocation).

Moving on to the next stratum, there are nine members where 'strata_group' is equal to "2". Counting out the previously described sequential pattern over the stratum (1,1,1,1,0,1,1,1,1), we would expect to see eight members assigned to "Study Arm 1" and one member assigned to "Study Arm 2". Once again as predicted, this is the case. However nine members is not a multiple of a CAU (5). The randomization approach applied by PROC SURVEYSELECT to this stratum has resulted in an 8:1 allocation. Finally there are nine members where 'strata_group' is equal to "3". Following the same logic; we would again expect, and do in fact find that eight members were assigned to "Study Arm 1" and one to "Study Arm 2", an 8:1 allocation.

Cumulatively, the twenty-three person study population has been randomized such that twenty members have been assigned to "Study Arm 1" and three to "Study Arm 2", resulting in a 6.7:1 allocation. The next step is to test that the strata means with respect to the primary outcome measure 'Score' are significantly different from one another, and seen by **Figure 1** this is the case. This is not a surprise as PROC SURVEYSELECT had no impact on the pre-assigned strata groups that were already clustered based on 'Score'.

Selected	strata_group	member	score	SelectionProb	SamplingWeight
1	1	a	0.1017	0.8	1.25
0	1	b	0.1082	0.8	0
1	1	c	0.1195	0.8	1.25
1	1	d	0.1071	0.8	1.25
1	1	e	0.1091	0.8	1.25
1	2	f	0.5178	0.888888889	1.125
0	2	g	0.5281	0.888888889	0
1	2	h	0.5152	0.888888889	1.125
1	2	i	0.5182	0.888888889	1.125
1	2	j	0.5091	0.888888889	1.125
1	2	k	0.5037	0.888888889	1.125
1	2	l	0.5117	0.888888889	1.125
1	2	m	0.5141	0.888888889	1.125
1	2	n	0.5015	0.888888889	1.125
1	3	o	0.9162	0.888888889	1.125
0	3	p	0.9218	0.888888889	0
1	3	q	0.9167	0.888888889	1.125
1	3	r	0.9122	0.888888889	1.125
1	3	s	0.9291	0.888888889	1.125
1	3	t	0.9114	0.888888889	1.125
1	3	u	0.9177	0.888888889	1.125
1	3	v	0.9219	0.888888889	1.125
1	3	w	0.9266	0.888888889	1.125

Table 1. Dataset "randomized"

Finally, the two Study Arm means (with respect to the same measure) should not be significantly different. That is the ratio of the primary outcome measures between "Study Arm 1" and "Study Arm 2", at the population level, should be as close to 1 as possible. As seen by **Figure 2** the study arm means are not significantly different, with a ratio equal to 1.1436.

```
  proc anova data = randomized;
   class strata_group;
   model score = strata_group;
   means strata_group / tukey; run;
   title1 'ANOVA Test on strata
   groupings'; run;


       ANOVA Test on strata Groupings

Comparisons  significant  at  the  0.05  level  are
indicated by ***.

            Difference
strata_group  Between    Simultaneous 95%
Comparison    Means      Confidence Limits

3 - 2      0.406022      0.397670  0.414375  ***
3 - 1      0.810169      0.800286  0.820052  ***
2 - 3     -0.406022     -0.414375 -0.397670  ***
2 - 1      0.404147      0.394264  0.414030  ***
1 - 3     -0.810169     -0.820052 -0.800286  ***
1 - 2     -0.404147     -0.414030 -0.394264  ***
```

```
  proc anova data = randomized;
   class Selected;
   model score = selected;
   means selected / tukey; run;
   title1 'ANOVA Test on Study Arms'; run;



        ANOVA Test on Study Arms

Means with the same letter are not
significantly different.

Tukey
Grouping  Mean      N    Selected

A       0.5940     20    1

A

A       0.5194      3    0
```

Figure 1. Testing the strata Means ✓   **Figure 2. Testing the Study Arm Means** ✓

The take away is that although the randomization passes statistical rigor, PROC SURVEYSELECT when randomizing a stratum population not equal to a perfect multiple of a CAU, will deviate from the selected allocation strategy. With a large enough population and a single randomization, this would likely be of little concern. Alternatively, when population numbers are small and/or study participants become eligible over multiple intervals (i.e. monthly), the potential for the unbalanced baseline outcome measures over the study arms is real.

RANDOMIZATION WITH REMAINDERS (SINGLE RANDOMIZATION EXAMPLE)

In the previous example, an ANOVA test proved that the population was stratified prior to randomization, in a manner that strata means (with respect to the measure 'Score') were significantly different from one another (**Figure 1**). We also saw that PROC SURVEYSELECT performed a stratified randomization such that the two resulting Study Arms had means (again with respect to the measure 'Score') that were not significantly different (**Figure 2**).

Still, the procedure was not successful with our third key component, staying true to the selected allocation strategy of 4:1 where the final allocation realized was 6.7:1. Can we employ an alternative approach to the previous example that results in Study Arms that are even more similar then those produced by PROC SURVEYSELECT and at the same time stay closer to the selected allocation strategy? What follows is exactly that.

The alternative solution that follows, 'Randomization with Remainders", has been written in macro form. Through three dynamic parameters it's able to meet the needs of most stratification approaches. I'll walk through each portion of the code starting with the macro parameters themselves.

```
%MACRO rand_remainders (strata_groups = , prop1 =  , fullclust =  );
```

The first parameter 'strata_groups' refers to the total number of strata within the population. For example, if a population was stratified on two variables, one with two levels (say sex) and another with three levels (say risk) the resulting number of strata would be "6". The parameter 'prop1' refers to the allocation strategy that is, it's asking for the proportion of records that should be assigned into Study Arm 1. In the example where the allocation was 4:1, this argument would be completed with ".80". Finally, 'fullclust' is asking for the number of records that make up a complete allocation unit (CAU). In the example of a 4:1 allocation, a complete allocation unit is "5".

```
proc sql;
  create table remainder_check as
  select strata_group,
         count(*) as new_members,
         count(*) / &fullclust. format=12.2 AS rand_ratio
  from to_be_randomized
  group by 1; quit:
```

The result of this section is a table with one record for each strata group. Each record describes the number of members within that stratum and a ratio of the member count (within that stratum) to the 'fullclust' parameter.

```
data remainder_check;
set remainder_check;
 new_rand_ratio = PUT(rand_ratio,12.2);
 remainder = SCAN(new_rand_ratio,2,".");
 if remainder = "00" Then heavy_by = 0;
 %dO r = 1 %tO (&fullclust. - 1);
 remainder_test = scan(put(&r. / &fullclust.,4.2),2,".");
 if remainder = remainder_test then heavy_by = &r.; %end;
 drop remainder_test; run;
```

The 'rand_ratio' created in the previous step, is converted into a character variable where the decimal now serves as a word delimiter. The character variable is scanned for the second word, which is translated into the quantity of members that exist within that stratum after the last complete allocation unit has been counted out, the remainders.

```
proc sql;
  select %do a = 1 %to &strata_groups.;
        (select heavy_by
         from remainder_check
         where strata_group = &a.), %end;
         %do b = 1 %tO &strata_groups.;
        (select new_members
         from remainder_check
         where strata_group = &b.)
         %if &b. ne  &strata_groups. %then %do;,%end; %end;
  into %do c = 1 %to &strata_groups.;
       :remainders_&c., %end;
       %do d = 1 %to &strata_groups.;
       :newmem_&d. %if &d. ne &strata_groups %then %do; , %end; %end;
  from remainder_check; quit:
```

```
%do e = 1 %to &strata_groups.;
%put "remainders_&e.: &&remainders_&e..";
%put "new_mem_&e.: &&newmem_&e.."; %end;
```

For each strata group the quantity of remainders and the quantity of members within that stratum are determined and placed into macro variables as well as written out to the log.

```
Proc sort data = to_be_randomized; by strata_group; run;
```

The dataset "to_be_randomized" is sorted by the 'strata_group' variable.

```
%do i = 1 %to &strata_groups.;
```

The code that follows is repeated a number of times equal to the quantity of strata groups as defined by the macro parameter "strata_group". Each iteration of the loop parses out and works on a single specific strata group.

```
data group_&i.;
 set to_be_randomized (where = (strata_group = &i.));
 sort_var = ranuni(0); run;

proc sort data = group_&i.; by sort_var; run;
```

For the strata group being worked on, a random uniform value over the interval (0,1) is created for each member. The stratum population is then sorted according to this new variable. This is the method for shuffling the member order within the stratum.

```
%let std_rand = %eval(&&newmem_&i. - &&remainders_&i.);
```

For the stratum being worked on, the quantity of members that will be placed through standard randomization is determined. This is equal to the total members within the group minus the remainders determined in an earlier step.

```
data group_&i._std_rand group_&i._man_rand;
set group_&i.;
 if _n_ le &std_rand. Then output group_&i._std_rand;
 if _n_ gt &std_rand. Then output group_&i._man_rand; run;
```

For the stratum being worked on, the randomly sorted members are placed into two datasets. The first dataset will contain a member count equal to the greatest whole multiple of a complete allocation unit, and will be put through standard randomization. The second dataset will contain the remaining members which will be randomized using an alternative method

Option 1: Manual

```
data group_&i._std_rand;
set group_&i._std_rand;
 count + 1;
 if count = (&fullclust. + 1)
 then count = 1;
 if count LT &fullclust.
 then study_arm = "Study Arm 1";
 if count = &fullclust.
 then study_arm = "Study Arm 2";
run;
```

Option 2: Proc SurveySelect

```
proc surveyselect data= group_&i._std_rand
 samprate = .80
 out = group_&i._std_rand
 method = SRS
 outall;
 strata strata_group; run;

data group_&i._std_rand;
set group_&i._std_rand;
 if selected = 1
 then study_arm = "Study Arm 1";
 else study_arm = "Study Arm 2";
 drop selected selectionprob samplingweight; run;
```

For the stratum being worked on I present two options. In **Option 1** the records within the standard randomization dataset are assigned a sequential value (1,2,3…) via the variable 'Count' up to the argument entered in the "fullclust" parameter, and repeated over all records. Because the quantity of records in this dataset is a multiple of a CAU, the last record will be assigned a count value equal to "fullclust". Those members assigned a count value less than the "fullclust" value are placed in "Study Arm 1" while records with a 'Count' value equal to "fullclust" are placed in "Study Arm 2". **Option 2** relies upon PROC SURVEYSELECT to perform the same task. Again, because the record count is a multiple of a CAU there will be no remainders. Both approaches stay true to the selected allocation strategy.

```
%if &&remainders_&i. NE 0 %then %do;
%do j = 1 %to &&remainders_&i.;

data temp_&i._&j.;
set group_&i._man_rand;
 if _n_ ne &j. then delete;
 random = ranuni(0);
 if random le &prop1. then study_arm = "Study Arm 1";
 if random gt &prop1. then study_arm = "Study Arm 2"; run; %end;

data group_&i._final;
set  group_&i._std_rand
     %do k = 1 %to &&remainders_&i.;
     temp_&i._&k. %end;; run; %end;

%if &&Remainders_&i. = 0 %then %do;
data group_&i._final;
set  group_&i._std_rand; run; %end; %end;
```

For the stratum being worked on, a new DO LOOP is run, with an iteration count equal to the number of remainders within that stratum. Each iteration of the loop focuses on the corresponding remainder within the dataset with the row number equal to the iteration number. That remainder is extracted from the "group_&i._man_rand" dataset and assigned a uniform random variable over the interval (0,1). When the value of the random variable less than or equal to the argument entered in the "prop1" parameter the member is assigned to "Study Arm 1", when the value is greater than "prop1" the member is assigned to "Study Arm 2". Each of the individual remainders (up to "fullclust" – 1) is brought back together with their arm assignments. In the absence of any remainders this process is bypassed.

Once complete, all members within that stratum are brought back together with their arm assignments. The program then proceeds to the next iteration, and runs through the randomization process for the next stratum.

```
data rand_profile;
set
%do q = 1 %to &strata_groups.;
    group_&q._final %end;; run;
```

At this point randomization is complete across all strata groups. All members across all strata groups are brought back together in a single dataset with their arm assignments.

```
%MEND rand_remainders;
%rand_remainders(strata_groups = 3, prop1 = .8, Fullclust = 5)
```

5

RANDOMIZATION WITH REMAINDERS RESULTS

The resulting dataset "rand_profile' (invoking *Option 1:* from page 4) is presented in **Table 2**. Once again, the twenty-three member study population has been randomized, only this time eighteen members have been assigned to "Study Arm 1" and five to "Study Arm 2", resulting in a 3.6:1 allocation.

As before, the strata means, with respect to the primary outcome measure 'Score', are significantly different from one another. This is expected as the same stratified member dataset was used in both approaches. Also we find the two Study Arm means (with respect to 'Score') are not significantly different from one another, with a ratio of 1.0218. Output of the two ANOVA tests are presented in **Figure 3** and **Figure 4** below.

As seen through the Proc SurveySelect method, the Randomization with Remainders approach passes statistical rigor. The difference here is that Randomization with Remainders results in a more attractive ratio of study arms and stays closer to the desired allocation. In the next section we explore a similar exercise on a hypothetical population being randomized over twelve monthly intervals.

member	strata_group	score	Sort_Var	Count	study_arm	Random
b	1	0.1082	0.161844	1	Study Arm 1	.
a	1	0.1017	0.442483	2	Study Arm 1	.
d	1	0.1071	0.619316	3	Study Arm 1	.
e	1	0.1091	0.645025	4	Study Arm 1	.
c	1	0.1195	0.966968	5	Study Arm 2	.
m	2	0.5141	0.069345	1	Study Arm 1	.
i	2	0.5182	0.231133	2	Study Arm 1	.
f	2	0.5178	0.271116	3	Study Arm 1	.
k	2	0.5037	0.505681	4	Study Arm 1	.
n	2	0.5015	0.568141	5	Study Arm 2	.
j	2	0.5091	0.634129	.	Study Arm 1	0.482736
g	2	0.5281	0.684198	.	Study Arm 1	0.560052
h	2	0.5152	0.874815	.	Study Arm 1	0.324598
l	2	0.5117	0.990237	.	Study Arm 2	0.931284
s	3	0.9291	0.11348	1	Study Arm 1	.
w	3	0.9266	0.122243	2	Study Arm 1	.
o	3	0.9162	0.32749	3	Study Arm 1	.
v	3	0.9219	0.433385	4	Study Arm 1	.
q	3	0.9167	0.448569	5	Study Arm 2	.
t	3	0.9114	0.674252	.	Study Arm 1	0.677522
u	3	0.9177	0.812299	.	Study Arm 1	0.195088
p	3	0.9218	0.835139	.	Study Arm 2	0.885596
r	3	0.9122	0.953954	.	Study Arm 1	0.084382

Table 2. Dataset "rand_profile"

```
  proc anova data = rand_profile;
   class strata_group;
   model score = strata_group;
   means strata_group / tukey; run;
  title1 'ANOVA Test on strata
  groupings'; run;


       ANOVA Test on strata Groupings

Comparisons significant at the 0.05 level are
indicated by ***.


             Difference
strata_group  Between    Simultaneous 95%
Comparison    Means      Confidence Limits

3 - 2     0.406022     0.397670  0.414375   ***
3 - 1     0.810169     0.800286  0.820052   ***
2 - 3    -0.406022    -0.414375 -0.397670   ***
2 - 1     0.404147     0.394264  0.414030   ***
1 - 3    -0.810169    -0.820052 -0.800286   ***
1 - 2    -0.404147    -0.414030 -0.394264   ***
```

Figure 3. Testing the strata Means ✓

```
  proc anova data = rand_profile;
   class study_arm;
   model score = study_arm;
   means study_arm / tukey; run;
  title1 'ANOVA Test on Study Arms'; run;


           ANOVA Test on Study Arms

Means with the same letter are not
significantly different.

Tukey
Grouping  Mean      N    study_arm

A       0.5942      5    Study Arm 1

A

A       0.5815     18    Study Arm 2
```

Figure 4. Testing the Study Arm Means ✓

6

MULTIPLE RANDOMIZATIONS EXAMPLE

In the previous example we explored the results of both methods when employed over a single randomization. Now we will investigate how these approaches compare when randomizing twelve populations over multiple intervals.

To prepare for this exercise twelve datasets were created, each containing twenty-three members. Each member was assigned to a strata group according to the following probabilities (in-line with the single randomization example): strata 1 (21.74%), strata 2 (39.13%) and strata 3 (39.13%). Once assigned to a strata group, each member was given a random 'Score' along the following intervals: strata 1 (.1000-.1999), strata 2 (.5000 - .5999) and strata 3 (.9000-.9999). The study members were then run through both randomization approaches and tested for statistical rigor. The results of the strata means test and study arm means test are present below by **Figures 5** and **6** and **Figures 7** and **8**. Again notice that the results of the strata means test (**Figures 5** and **7**) are identical as both approaches used the same stratified datasets. Additionally, the two approaches produced study arms with non-significantly different study arm means. The code to develop these datasets has been attached in APPENDIX 1.

PROC SURVEYSELECT

```
 proc anova data = randomized;
  class strata_group;
  model score = strata_group;
  means strata_group / tukey; run;
  title1 'ANOVA Test on strata
  Groupings'; run;

      ANOVA Test on strata Groupings

Comparisons significant at the 0.05 level are
indicated by ***.

           Difference
strata_group Between   Simultaneous 95%
Comparison   Means     Confidence Limits

3 - 2     0.395818     0.386921  0.404715  ***
3 - 1     0.798448     0.787772  0.809123  ***
2 - 3    -0.395818    -0.404715 -0.386921  ***
2 - 1     0.402630     0.391955  0.413305  ***
1 - 3    -0.798448    -0.809123 -0.787772  ***
1 - 2    -0.402630    -0.413305 -0.391955  ***
```

```
 proc anova data = randomized;
  class selected;
  model score = selected;
  means selected / tukey; run;
  title1 'ANOVA Test on Study Arms';  run;

          ANOVA Test on Study Arms

Means with the same letter are not
significantly different.

Tukey
Grouping  Mean       N     Selected

A      0.63546    41    0

A

A      0.61909   235    1
```

Figure 5. Testing the strata Means ✓   **Figure 6. Testing the Study Arm Means** ✓

Randomization with Remainders

```
 proc anova data = randomized;
  class strata_group;
  model score = strata_group;
  means strata_group / tukey; run;
  title1 'ANOVA Test on strata
  Groupings'; run;

      ANOVA Test on strata Groupings

Comparisons significant at the 0.05 level are
indicated by ***.

           Difference
strata_group Between   Simultaneous 95%
Comparison   Means     Confidence Limits

3 - 2     0.395818     0.386921  0.404715  ***
3 - 1     0.798448     0.787772  0.809123  ***
2 - 3    -0.395818    -0.404715 -0.386921  ***
2 - 1     0.402630     0.391955  0.413305  ***
1 - 3    -0.798448    -0.809123 -0.787772  ***
1 - 2    -0.402630    -0.413305 -0.391955  ***
```

```
 proc anova data = rand_profile;
  class study_arm;
  model score = study_arm;
  means study_arm / tukey; run;
  title1 'ANOVA Test on Study Arms'; run;

          ANOVA Test on Study Arms

Means with the same letter are not
significantly different.

Tukey
Grouping  Mean       N    study_arm

A      0.62209   224   Study Arm 1

A

A      0.61909    52   Study Arm 2
```

Figure 7. Testing the strata Means ✓   **Figure 8. Testing the Study Arm Means** ✓

SUMMARY RESULTS

Single Randomization Example

Table 3 summarizes the stratified random sampling results of the "Single Randomization" examples. In both approaches the same initial member dataset was used and run through each respective method. This accounts for why both approaches result in the same number of strata mean pairings significantly different from one another. Most important, both techniques passed the statistical tests to ensure that the study arm means (with respect to the primary outcome measure 'Score') were not significantly different from one another, the core objective of random sampling. Still the results were very different.

In each category "Randomization with Remainders" outperformed PROC SURVEYSELECT. That is with "Randomization with Remainders", the baseline primary outcome measure and the final membership allocation were both closer to the ideal outcomes of "1" and "4:1" respectively.

Randomization Approach	strata Pairings Significantly Different	Ratio of Means (Study Arm 1 / Study Arm 2)	Percent Away from an Ideal Ratio of 1	Allocation as a Result of Method	Absolute Difference from the Specified Allocation as a Percentage
Randomization with Remainders	All	0.9786	2.14%	3.60:1	10.00%
Proc SurveySelect	All	1.1436	14.36%	6.67:1	66.67%

Table 3. Comparison of Approaches (Single Randomization)

Multiple Randomizations Example

Table 4 summarizes the stratified random sampling results of the "Multiple Randomizations" examples. In both approaches the same twelve member datasets were used and run through each respective method. This accounts for why both approaches result in the same number of strata mean pairings significantly different from one another. Most important, both techniques passed the statistical tests to ensure that the study arm means (with respect to the primary outcome measure 'Score') were not significantly different from one another, the core objective of random sampling. Once again, the results were very different.

Both methods improved their performance compared to the first, single randomization example. Still, "Randomization with Remainders" outperformed PROC SURVEYSELECT in every category. That is with "Randomization with Remainders", the ratio of the study arm means (with respect to the primary outcome measure 'Score') and the final membership allocation were closer to the ideal outcomes of "1" and "4:1" respectively.

Randomization Approach	strata Pairings Significantly Different	Ratio of Means (Study Arm 1 / Study Arm 2)	Percent Away from an Ideal Ratio of 1	Allocation as a Result of Method	Absolute Difference from the Specified Allocation as a Percentage
Randomization with Remainders	All	1.0048	0.48%	4.31:1	7.69%
Proc SurveySelect	All	0.9742	2.58%	5.73:1	43.29%

Table 4. Comparison of Approaches (Multiple Randomizations)

CONCLUSION

When stratified populations are large, PROC SURVEYSELECT does a satisfactory job in randomizing the sub populations. Within any stratum, the desired proportion will be out of balance by a member count no greater than a complete allocation unit minus one. It is when population numbers are small and/or study participants become eligible over multiple intervals (i.e. monthly), that flags are raised. Over time this has the potential for allowing bias into the study, where one study arm is likely to exceed its targeted proportion, heavy with records from lightly populated stratum. Even worse, this may lead to the imbalance of baseline measures of interest.

"Randomization with Remainders" at worse, performs on par with PROC SURVEYSELECT but often times better. Randomization with Remainders identifies the quantity of members beyond the last multiple of a complete allocation unit and randomizes them individually. Each remainder is then randomized into a study arm with a probability that is in-line with the targeted allocation strategy. This has the effect of staying closer to the true allocation strategy and producing a ratio of baseline study arm means that is closer to the ideal outcome "1", implying perfect balance.

CONTACT INFORMATION

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

Name: Gregg Oginz
Enterprise: Health Dialog
Address: 6 Bedford Farms
City, State ZIP: Bedford, NH 03106
Work Phone: 603.628.5132
Linked in: www.linkedin.com/in/greggoginz
E-mail: goginz@healthdialog.com
Web: www.healthdialog.com

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 1: MONTHLY INTERVAL CODE

```
data members;
input study_member $;
datalines;
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
q
r
s
t
u
v
w
; run;

/*Dataset development and Proc SurveySelect Approach*/
%Macro members;
%do i = 1 %to 12;
data members_&i.;
set members;
 member = COMPRESS(Study_member||"_&i.");
 Shuffle = ranuni(0);
 if shuffle le .2174 then strata_group = 1;
 if .2174 LT shuffle le .6087 then strata_group = 2;
 if shuffle gt .6087 then strata_group = 3;
 if strata_group = 1 then score = (ranuni(0)/10)+.1;
 if strata_group = 2 then score = (ranuni(0)/10)+.5;
 if strata_group = 3 then score = (ranuni(0)/10)+.9;
 drop study_member; run;

proc sort data= members_&i.; by strata_group; run;

proc surveyselect data = members_&i.
 samprate = .80
 out = randomized_&i.
 method = SRS
 outall;
 strata strata_group; %end; run;

data all_members;
 set %do j = 1 %to 12;
     randomized_&j. %end;; run;

proc sort data= all_members; by strata_group; run;

%MEND members;
%members;
```

```
/*Randomization with Remainders Approach*/
%MACRO rand_remainders(strata_groups = , prop1 =  , fullclust =  );

%do v = 1 %to 12;
proc sql noprint;
 create table remainder_check as
 select strata_group,
        cOUNT(*) as new_members,
        cOUNT(*) / &fullclust. format=12.2 as rand_ratio
 from members_&v.
 group by 1; quit:

data remainder_check;
 set remainder_check;
 new_rand_ratio = put(rand_ratio,12.2);
 remainder = scan(new_rand_ratio,2,".");
 if remainder = "00" Then heavy_by = 0;
 %do r = 1 %to (&fullclust. - 1);
 remainder_test = scan(put(&r. / &fullclust.,4.2),2,".");
 if remainder = remainder_test then heavy_by = &r.; %end;
 drop remainder_test; run;

proc sql noprint;
 select %do a = 1 %to &strata_groups.;
       (select heavy_by
        from remainder_check
        where strata_group = &a.), %end;
        %do b = 1 %to &strata_groups.;
       (select new_members
        from remainder_check
        where strata_group = &b.)
        %if &b. ne  &strata_groups. %then %do;,%end; %end;
 into %do c = 1 %to &strata_groups.;
      :remainders_&c., %end;
      %do d = 1 %to &strata_groups.;
      :newmem_&d. %if &d. ne &strata_groups %then %do; , %end; %end;
 from remainder_check; quit:

%do e = 1 %to &strata_groups.;
%put "remainders_&e.: &&remainders_&e..";
%put "new_mem_&e.: &&newmem_&e.."; %end;

proc sort data = members_&v.; by strata_group; run;

%do i = 1 %to &strata_groups.;

data group_&i.;
set members_&v. (where = (strata_group = &i.));
 sort_var = ranuni(0); run;

proc sort data = group_&i.; by sort_var; run;

%let std_rand = %eval(&&newmem_&i. - &&remainders_&i.);

data group_&i._std_rand group_&i._man_rand;
set group_&i.;
 if _n_ le &std_rand. then output group_&i._std_rand;
 if _n_ gt &std_rand. then output group_&i._man_rand; run;
```

```
data group_&i._std_rand;
set group_&i._std_rand;
 count + 1;
 if count = (&fullclust. + 1) then count = 1;
 if count lt &fullclust.        then study_arm = "Study Arm 1";
 if count =  &fullclust.        then study_arm = "Study Arm 2"; run;

%if &&remainders_&i. ne 0 %then %do;
%do j = 1 %to &&remainders_&i.;

data temp_&i._&j.;
set group_&i._man_rand;
 if _N_ NE &j. then delete;
 random = ranuni(0);
 if random le &prop1. then study_arm = "Study Arm 1";
 if random gt &prop1. then study_arm = "Study Arm 2"; run; %end;

data group_&i._final;
set  group_&i._std_rand
     %do k = 1 %to &&remainders_&i.;
     temp_&i._&k. %end;; run; %end;

%if &&remainders_&i. = 0 %then %do;
data group_&i._final;
set  group_&i._std_rand; run; %end;

%end;

data rand_profile_&v.;
set %do q = 1 %to &strata_groups.;
    group_&q._final %end;; run;

%end;

data rand_profile;
set %do y = 1 %to 12;
    rand_profile_&y. %end;; run;

%MEND rand_remainders;
%rand_remainders(strata_groups = 3, prop1 = .8, Fullclust = 5)
```