Chapter 269

Randomization Lists

Introduction

This module is used to create a randomization list for assigning subjects to one of up to eight groups or treatments. Six randomization algorithms are available. Four of the algorithms (Efron's biased coin randomization, Smith's randomization, Wei's urn randomization, and random sorting using maximum allowable % deviation) are designed to generate balanced random samples throughout the course of an experiment. The other two (complete randomization and random sorting) are less complex but have higher probabilities of imbalance. Technical details for each of these algorithms (except random sorting) as well as a discussion of randomization issues in general are given in Rosenberger and Lachin (2002), Pocock (1983), and Piantadosi (2005).

Why Randomize?

Random allocation has been an important part of scientific discovery since the time of R.A. Fisher. He and other scientists recognized the potential for bias in nonrandomized experiments that could compromise treatment comparisons. Randomization controls the influence of unknown factors, reducing potential bias in an experiment. Rosenberger and Lachin (2002) state on page 18 that "…the randomization of subjects between the treatment groups is the paramount statistical element that allows one to claim that a study is unbiased." The elimination of bias, however, is not the only reason for randomization. Pocock (1983) states on page 50 that randomization is used "to provide a basis for the standard methods of statistical analysis such as significance tests." Clearly, randomization is necessary in order to produce accurate and generalizable statistical conclusions.

Randomization in Clinical Trials

In non-clinical (or non-human) experiments, researchers often have tight control over the environment and conditions surrounding an experiment, so randomization can usually be implemented with minor difficulty. Clinical experiments, however, are quite different. Because human patients are used in clinical trials, various ethical issues must be addressed. These ethical considerations complicate the experimental design and often require adjustments in the way subjects are randomly assigned to treatments. Other factors that influence randomization in clinical trials are purely logistical. How can the investigators ensure that treatments are administered the same for all patients without the patient or the doctor knowing what treatment is being given? These and other design factors influence how randomization is administered in clinical trials.

One of the issues that arise in clinical experiments is treatment imbalance. Clinical trials are usually administered over time with patients enrolling at different time points throughout the study. It is often desirable to maintain balance in the number of patients assigned to each treatment throughout the course of the experiment. This is particularly true when time-dependent covariates influence the response or when sequential testing will be used to analyze results. Several randomization algorithms have been developed to produce lists that balance the number of patients assigned to each treatment throughout the experiment while still maintaining the randomness of the assignments. These include Efron's biased coin randomization, Smith's randomization, Wei's urn randomization, and random sorting using maximum allowable % deviation. Each of these algorithms will be discussed in detail in the section that follows.

Randomization Algorithms

Six different randomization algorithms are available in ***NCSS***. These can be roughly divided into two categories: those that aim to produce balanced randomization lists and those that do not. The following table outlines the goal of each algorithm by the number of groups each algorithm will allow.

	Non-Balancing Algorithms	**Balancing Algorithms**
2 Groups	Complete Randomization†, Random Sorting	Efron's Biased Coin*†, Smith*†, Wei's Urn*†, Random Sorting using Maximum Allowable % Deviation
***k* Groups**	Complete Randomization†, Random Sorting	Wei's Urn*, Random Sorting using Maximum Allowable % Deviation

*These randomization algorithms have the additional restriction that unequal treatment allocation is not allowed, i.e. all groups must have the same target sample size.

†These randomization algorithms produce randomization lists in which the actual group sample size may not equal the target group sample size.

The discussion of each algorithm that follows will be based on the following scenario and notation. Suppose we have k treatments and that n_i subjects (not necessarily all equal) are to be assigned to each treatment, i = 1, 2, …, k. The value n_i will be referred to in discussion as the "target" sample size for each group. Let the actual sample size for each group be a_i. For some algorithms, the actual group sample size may not always equal the target sample size for all groups. The total sample size is

$$N = \sum_{i=1}^{k} n_i = \sum_{i=1}^{k} a_i \ ,$$

and the target allocation ratio for each group is

$$R_i = \frac{n_i}{N} \, .$$

Define the probability of assignment of subject j to treatment i as p_{ij}.

Define the number of subjects in each group after the j^{th} subject is assigned as $n_i[j]$.

Non-Balancing Algorithms

Complete Randomization

The complete randomization algorithm is commonly referred to as a "coin flip". For the case to two treatments, a coin is flipped each time a subject is to be randomized, determining the assignment. The probability of assignment to either treatment is 0.5 for all subjects.

The algorithm generalizes to more than two groups and allows for unequal allocation. The probability of assignment of subject j to group i is

$$p_{ij} = R_i$$

for all j. If the target sample size is the same for all k groups, then

$$p_{ij} = 1/k$$

The complete randomization algorithm proceeds by randomly assigning subjects to treatments using the above assignment probabilities. The algorithm may result in imbalances between groups even when the target group sample sizes are equal, i.e. the actual sample sizes may not always equal the target sample sizes for all groups. This algorithm is not recommended when balance is important throughout the course of an experiment.

Random Sorting

The random sorting algorithm can be used for any number of treatment groups and any allocation ratios. The random sorting algorithm always results in randomized assignment lists in which the actual group sample sizes match the target group sample sizes, i.e. $a_i = n_i$ for all i. The algorithm begins by creating a virtual database containing group names. Each row corresponds to one group, and the i^{th} group is represented by n_i rows. For example, if we were randomly assigning three groups (A, B, and C) with ten subjects in each group, then the database would consist of ten rows containing A's, followed by ten rows containing B's and ten rows containing C's, for a total of 30 rows. Next, a random number is assigned to each row in the database, and then the database is sorted based on the column of random numbers to place the group names in random order. The first subject is then assigned to the group in row one of the randomly sorted database, the second subject is assigned to the group in row two, and so forth. Although this algorithm always results in groups containing the target sample sizes, longitudinal imbalances among groups may still occur, therefore, this algorithm is not recommended when balance is important throughout the course of an experiment.

Balancing Algorithms

Efron's Biased Coin Randomization

This algorithm may only be used for random assignment of subjects to two treatments. The target sample sizes must also be the same for both groups. In order to achieve longitudinal balance between groups, the algorithm dynamically changes the group assignment probabilities. The algorithm is outlined in Efron (1971).

First we define a constant probability p (called "Efron's p" in ***NCSS***), where $0.5 < p \leq 1$. A common value for p is 2/3. Also define a difference function $D_j = n_1[j] - n_2[j]$. The probability of assignment of subject j to group 1 is

$$p_{1j} = \begin{cases} 1/2 & \text{if} \quad D_{j-1} = 0 \\ p & \text{if} \quad D_{j-1} < 0 \\ 1-p & \text{if} \quad D_{j-1} > 0 \end{cases}$$

Efron's biased coin randomization proceeds by randomly assigning subjects to treatments using the above assignment probabilities. When group 1 has more subjects assigned than group 2, the assignment probability changes to make group 2 more probable for assignment. When group 2 has more, then group 1 becomes more probable. The algorithm may result in final imbalances between groups, but the degree of imbalance throughout the randomization process is greatly reduced.

Smith's Randomization

This algorithm may only be used for random assignment of subjects to two treatments. The target sample sizes must also be the same for both groups. Like Efron's biased coin randomization, Smith's algorithm dynamically changes the group assignment probabilities based on the degree of imbalance to achieve longitudinal balance between groups. The algorithm is outlined in Smith (1984).

We define a positive exponent parameter ρ (called "Smith's Exponent" in ***NCSS***). The probability of assignment of subject j to group 1 is

$$p_{1j} = \frac{n_2[j-1]^\rho}{n_1[j-1]^\rho + n_2[j-1]^\rho}$$

Smith's randomization proceeds by randomly assigning subjects to treatments using the above assignment probabilities. When group 1 has more subjects assigned than group 2, the assignment probability changes to make group 2 more probable for assignment. When group 2 has more, then group 1 becomes more probable. The algorithm may result in final imbalances between groups, but the degree of imbalance throughout the randomization process is greatly reduced.

Wei's Urn Randomization

This algorithm may be used for random assignment of subjects to two or more treatments. The target sample sizes must also be the same for all groups. Like Smith's randomization, Wei's urn randomization algorithm dynamically changes the group assignment probabilities based on the degree of imbalance to achieve longitudinal balance between groups. Urn randomization is reviewed in Wei and Lachin (1988).

Define positive parameters *A* and *B* (called "Wei's A" and "Wei's B" in ***NCSS***, respectively). We start the algorithm by placing *A* balls representing each group in an urn. A single ball is then randomly chosen from the urn, recorded, and *replaced*, and then *B* new balls corresponding to each of the other groups are added to the urn. Therefore, when a ball from one group is chosen, the probability shifts to make the other groups more probable on the next draw. The probability of the first assignment is $1/k$. After that, the probability of assignment of subject j to group i is

$$p_{ij} = \frac{A + B(j-1) - Bn_i[j-1]}{kA + B(j-1)(k-1)}.$$

The algorithm continues until all subjects have been assigned to one of the groups. The algorithm may result in final imbalances between groups, but the degree of imbalance throughout the randomization process is diminished due to the shifting of probabilities toward the underrepresented groups.

Random Sorting using Maximum Allowable % Deviation

This algorithm is equivalent to the random sorting algorithm described earlier except that a search is conducted to find a randomization list that satisfies the Maximum Allowable % Deviation criterion. The % Deviation for group i after subject j has been assigned is defined as

$$\%\ \text{Deviation}_{ij} = \left|\frac{n_i[j] - E(n_i[j])}{n_i}\right| \times 100$$

$$= \left|\frac{n_i[j] - jR_i}{n_i}\right| \times 100.$$

The % Deviation measures how far the actual sample size for group i is from the expected sample size after subject j is randomly assigned. The Maximum Allowable % Deviation represents the upper bound for this measure. The search is conducted by creating an assignment list based on random sorting and then running through the assignments and calculating the maximum % Deviation for all groups after each assignment. If the maximum % Deviation is greater than the Maximum Allowable % Deviation value specified, then the list is rejected, the number of iterations is incremented, and the random sorting algorithm is started again with a new set of random numbers. The search continues until a randomization list is generated for which the criterion is satisfied for all individual assignments. Conducting a search in this manner assures a degree of balance throughout the course of the experiment.

For example, for 40 subjects to be assigned to two groups A and B with equal allocation ratios (0.5), suppose that there are 7 assigned A's and 3 assigned B's after 10 random assignments and the Maximum Allowable % Deviation is 10%. With the allocation ratio at 0.5, we would expect to have 5 A's and 5 B's (10 × 0.5 = 5) after 10 assignments. Therefore, the % Deviation for group A is |7 – 5|/20 = 10% and the % Deviation for group B is |3 – 5|/20 = 10%. Both of these are equal to the Maximum Allowable % Deviation so the next assignment would be tested. If the next assignment were to group A then the randomization list would be rejected because the %

Deviation for group A is |8 – 5|/20 = 15% > 10%. A new randomization list based on random sorting would be generated and the search would continue.

Comparison of Balancing Properties

Rosenberger and Lachin (2002) provides a simulation comparison of the balancing properties of complete randomization, Efron's biased coin (p = 2/3), Wei's Urn (A = 0, B = 1), and Smith (exponent = 5). The simulation was carried out for two treatment groups with target sample sizes of 25 in each group. They found that complete randomization did not balance as well as the other three "restricted randomization" procedures. Efron's biased coin and Smith's randomization algorithms were very close in terms of bias and variability. Wei's urn was found to be slightly more variable.

Procedure Options

This section describes the options that are specific to this procedure. These are located on the Design, Report, and Storage tabs. For more information about the options of other tabs, go to the Procedure Window chapter.

Design Tab

The Design tab contains most of the parameters and options that you will be concerned with.

Randomization Algorithm

Randomization Algorithm

Choose one of six randomization algorithms. These algorithms are each described in detail in the Randomization Algorithms section above. Some of the algorithms may result in actual (final) sample sizes that are not equal to the target sample sizes. To search for a randomization list in which the final group sizes are equal to the target group sample sizes for every group, check the "Search for a randomization list…" box.

The following algorithms are available.

- **Complete Randomization**
  This is the simplest of all randomization algorithms and is commonly known as "coin flipping". The algorithm may result in imbalances between groups even when the target group sample sizes are equal, i.e. the actual sample sizes may not always equal the target sample sizes for all groups. This algorithm is not recommended when balance is important throughout the course of an experiment.

- **Efron's Biased Coin (2 Treatments, Equal Sample Sizes)**
  This algorithm may only be used for random assignment of subjects to two treatments. The target sample sizes must also be the same for both groups. In order to achieve longitudinal balance between groups, the algorithm dynamically changes the group assignment probabilities. A value for Efron's p must also be specified.

- **Random Sorting**
  The random sorting algorithm can be used for any number of treatment groups and any allocation ratio set. The random sorting algorithm always results in randomized assignment lists in which the actual group sample sizes match the target group sample sizes, i.e. $a_i = n_i$ for all i. The random assignment is conducted by sorting a database of group labels using a column of random numbers. This algorithm is not recommended when balance is important throughout the course of an experiment.

- **Random Sorting using Max Allowable % Deviation (Search)**
  This algorithm is equivalent to the random sorting algorithm except that a search is conducted to find a randomized list for which all % Deviations are less than or equal to the Maximum Allowable % Deviation specified. The search is conducted until a list is found or the maximum iterations are reached. A value for Maximum Allowable % Deviation must also be specified.

- **Smith (2 Treatments, Equal Sample Sizes)**
  This algorithm may only be used for random assignment of subjects to two treatments. The target sample sizes must also be the same for both groups. Smith's algorithm dynamically changes the group assignment probabilities based on the degree of imbalance to achieve longitudinal balance between groups. A value for Smith's Exponent must also be specified.

- **Wei's Urn (Equal Sample Sizes)**
  This algorithm may be used for random assignment of subjects to two or more treatments. The target sample sizes must also be the same for all groups. Wei's urn randomization algorithm dynamically changes the group assignment probabilities based on the degree of imbalance to achieve longitudinal balance between groups. Values for Wei's A and Wei's B must also be specified.

Randomization Algorithm – Algorithm Parameters

Efron's p

Specify the probability parameter *p* for Efron's bias coin algorithm. This value must be greater than 0.5 and less than or equal to 1. A probability of 0.67 is often used.

Maximum Allowable % Deviation

Specify a single value for the Maximum Allowable % Deviation for the random sorting using maximum allowable % deviation algorithm. You can enter the value as a percentage (e.g. 10%) or as a decimal (e.g. 0.10).

Smith's Exponent

Specify the exponent parameter ρ for Smith's randomization algorithm. Smith (1984) favors the design with ρ = 5.

Wei's A

Specify the initialization parameter *A* for Wei's urn algorithm. This is the number of balls in the urn for each group when the algorithm starts. A value of 0 is often used.

Wei's B

Specify the parameter *B* for Wei's urn algorithm. This is the number of balls added to the other urns when a ball is selected. A value of 1 is often used. If *B* is 0 then Wei's algorithm is the same as complete randomization.

Randomization Algorithm – Search

Search for a randomization list in which all final group sizes exactly match the target group sizes

Some of the randomization algorithms may result is actual group sample sizes that do not exactly match the target group sample sizes specified. This option forces a search to find a randomization list in which the final sample sizes match the target sample sizes for all groups.

The search is conducted by creating a randomization list using the user-specified randomization algorithm and then looking at the final sample sizes. If the sample sizes do not match the target sample sizes for all groups, then the list is discarded and the algorithm is restarted. This continues until a list with the exact sample sizes is found.

Maximum Iterations in Search

Specify the number of iterations before the randomization list search is terminated. This option is used for the Random Sorting using Max Allowable % Deviation (Search) algorithm and in searching for a randomization list where final group sizes match the target group sizes.

Specify the Groups

Number of Groups

This option specifies the number of groups or treatments to which the subjects will be randomly assigned. Up to eight groups are allowed for random assignment.

Specify the Groups – Group Labels and Target Group Sizes

All Group Sizes are Equal

Check this box if all groups have the same target sample size. The target sample size is then specified in the Equal Group Size box to the right. This is used as the target sample size for every group. If this box is not checked, then a target sample size must be entered for each group individually.

Equal Group Size

Specify a single target sample size to be used for all of the groups. This box is only used if the All Group Sizes are Equal option is checked. If the target sample sizes are not all equal, then they must be entered individually for each group.

Group Label

Specify a label or treatment name for each group. This label will be used to distinguish groups in the output and reports. Each group must have a different label.

Group Size

Specify a target sample size for the group designated to the left of this box. This option is only used if the All Group Sizes are Equal box is unchecked.

If all of the target group sizes are equal, check the All Group Sizes are Equal box and enter a single number in the Equal Group Size box.

Storage Tab

The storage tab contains options for specifying data storage columns.

Data Storage

Store Randomization List

If this box is checked, then the randomization list will be written to the data table using the specified columns. Writing the list to the data table allows you to easily store the randomization list for further analysis in ***NCSS***.

Store Subject ID's In

Specify a data column in which to store the subject ID values.

Store Group Assignments In

Specify a data column in which to store the randomized group assignments.

Reports Tab

The Reports tab contains options influencing the reports.

Subject ID Prefixes

Subject ID Prefix

Enter a text value to be added to the beginning of each subject ID. For example, if this value were "sub_" then the subjects would be labeled sub_1, sub_2, sub_3, etc. in the reports.

Example 1 – Randomization with Equal Allocation Ratios

A clinical researcher wishes to randomly assign 30 subjects to three treatments. The requirement is that each group has exactly 10 subjects and that the maximum % deviation is no larger than 20% of the target sample size.

Setup

This section presents the values of each of the parameters needed to run this example. First, from the data window, load the **Randomization Lists** procedure window using the Analysis menu or the Procedure Navigator. You may then make the appropriate entries as listed below, or open **Example 1** by going to the **File** menu and choosing **Open Example Template**.

Option	Value
Design Tab	
Randomization Algorithm	**Random Sorting using Max Allowable % Deviation**
Maximum Allowable % Deviation	**20%**
Search for exact match	**Checked**
Maximum Iterations in Search	**1000**
Number of Groups	**3**
All Group Sizes are Equal	**Checked**
Equal Group Size	**10**
Grp 1 Label	**A**
Grp 2 Label	**B**
Grp 3 Label	**C**

Annotated Output

Click the Run button to perform the calculations and generate the following output. Your output will be different because of random assignment.

Summary and References Sections

Summary		
Randomization Algorithm	Random Sorting (Maximum Allowable % Deviation = 20%)	
Search Iterations	4	
Number of Groups	3	
Total Sample Size	30	
Group Sample Sizes	Actual	Target
-- A	10	10
-- B	10	10
-- C	10	10

References
Piantadosi, S. 2005. Clinical Trials - A Methodological Perspective. John Wiley & Sons. New Jersey.
Pocock, S.J. 1983. Clinical Trials - A Practical Approach. John Wiley & Sons. New York.
Rosenberger, W.F., and Lachin, J.M. 2002. Randomization in Clinical Trials - Theory and Practice. John Wiley & Sons. New York.

This report displays the summary of the randomization algorithm and references.

Randomization List Section

Randomization List

Subject ID	Group Assignment	Largest % Deviation from Target	Cumulative Sample Size (A, B, C)
1	C	6.7%	(0, 0, 1)
2	A	6.7%	(1, 0, 1)
3	B	0.0%	(1, 1, 1)
4	C	6.7%	(1, 1, 2)
5	C	13.3%	(1, 1, 3)
6	A	10.0%	(2, 1, 3)
7	C	16.7%	(2, 1, 4)
8	B	13.3%	(2, 2, 4)
9	A	10.0%	(3, 2, 4)
10	B	6.7%	(3, 3, 4)
11	A	6.7%	(4, 3, 4)
12	A	10.0%	(5, 3, 4)
13	A	16.7%	(6, 3, 4)
14	C	16.7%	(6, 3, 5)
15	B	10.0%	(6, 4, 5)
16	B	6.7%	(6, 5, 5)
17	C	6.7%	(6, 5, 6)
18	C	10.0%	(6, 5, 7)
19	B	6.7%	(6, 6, 7)
20	A	6.7%	(7, 6, 7)
21	C	10.0%	(7, 6, 8)
22	A	13.3%	(8, 6, 8)
23	A	16.7%	(9, 6, 8)
24	B	10.0%	(9, 7, 8)
25	C	13.3%	(9, 7, 9)
26	B	6.7%	(9, 8, 9)
27	C	10.0%	(9, 8, 10)
28	B	6.7%	(9, 9, 10)
29	A	6.7%	(10, 9, 10)
30	B	0.0%	(10, 10, 10)

This report shows the complete randomization list with details after each assignment. The largest observed % deviation from the target is 16.7%.

Subject ID

The identification value of the current subject.

Group Assignment

The group to which the current subject was randomly assigned.

Largest % Deviation from Target

The largest observed % deviation after the current assignment was made. This measures how far away the group sample sizes are from the expected sample size based on the targets.

Cumulative Sample Size (Grp 1, Grp 2, …, Grp 8)

The cumulative sample size total for each group after the current assignment was made.

Example 2 – Randomization with Unequal Allocation Ratios

A researcher wishes to randomly assign 40 subjects to three treatments using complete randomization. The requirement is that first group has exactly 20 subjects and the other two have exactly 10 subjects each.

Setup

This section presents the values of each of the parameters needed to run this example. First, from the data window, load the **Randomization Lists** procedure window using the Analysis menu or the Procedure Navigator. You may then make the appropriate entries as listed below, or open **Example 2** by going to the **File** menu and choosing **Open Example Template**.

Option	Value
Design Tab	
Randomization Algorithm	**Complete Randomization**
Search for exact match …	**Checked**
Maximum Iterations in Search	**1000**
Number of Groups	**3**
All Group Sizes are Equal	**Unchecked**
Grp 1 Label	**Control**
Grp 1 Size	**20**
Grp 2 Label	**A**
Grp 2 Size	**10**
Grp 3 Label	**B**
Grp 3 Size	**10**

Output

Click the Run button to perform the calculations and generate the following output. Your output will be different because of random assignment.

Results

Summary

Randomization Algorithm	Complete Randomization	
Search Iterations	66	
Number of Groups	3	
Total Sample Size	40	
Group Sample Sizes	Actual	Target
-- Control	20	20
-- A	10	10
-- B	10	10

Randomization List

Subject ID	Group Assignment	Largest % Deviation from Target	Cumulative Sample Size (Control, A, B)
1	A	7.5%	(0, 1, 0)
2	B	5.0%	(0, 1, 1)
3	B	12.5%	(0, 1, 2)
4	Control	10.0%	(1, 1, 2)
5	Control	7.5%	(2, 1, 2)
6	Control	5.0%	(3, 1, 2)
7	Control	7.5%	(4, 1, 2)
8	A	0.0%	(4, 2, 2)
9	B	7.5%	(4, 2, 3)
10	B	15.0%	(4, 2, 4)
11	B	22.5%	(4, 2, 5)
12	B	30.0%	(4, 2, 6)
13	Control	27.5%	(5, 2, 6)
14	Control	25.0%	(6, 2, 6)
15	B	32.5%	(6, 2, 7)
16	Control	30.0%	(7, 2, 7)
17	A	27.5%	(7, 3, 7)
18	B	35.0%	(7, 3, 8)
19	A	32.5%	(7, 4, 8)
20	Control	30.0%	(8, 4, 8)
21	Control	27.5%	(9, 4, 8)
22	A	25.0%	(9, 5, 8)
23	A	22.5%	(9, 6, 8)
24	A	20.0%	(9, 7, 8)
25	Control	17.5%	(10, 7, 8)
26	Control	15.0%	(11, 7, 8)
27	B	22.5%	(11, 7, 9)
28	Control	20.0%	(12, 7, 9)
29	Control	17.5%	(13, 7, 9)
30	Control	15.0%	(14, 7, 9)
31	Control	12.5%	(15, 7, 9)
32	Control	10.0%	(16, 7, 9)
33	A	7.5%	(16, 8, 9)
34	B	15.0%	(16, 8, 10)
35	Control	12.5%	(17, 8, 10)
36	Control	10.0%	(18, 8, 10)
37	Control	12.5%	(19, 8, 10)
38	A	5.0%	(19, 9, 10)
39	A	2.5%	(19, 10, 10)
40	Control	0.0%	(20, 10, 10)

The algorithm required 66 iterations to find a list using complete randomization in which the target and actual sample sizes are equal.

Example 3 – Saving the Randomization List to the Spreadsheet

A researcher would like to randomize 20 subjects to two groups using Efron's biased coin randomization and save the randomization list to the spreadsheet. The researcher is targeting 10 per group, but will allow imbalance if it results from the algorithm.

Setup

This section presents the values of each of the parameters needed to run this example. First, from the data window, load the **Randomization Lists** procedure window using the Analysis menu or the Procedure Navigator. You may then make the appropriate entries as listed below, or open **Example 3** by going to the **File** menu and choosing **Open Example Template**.

Option	Value
Design Tab	
Randomization Algorithm	**Efron's Biased Coin**
Efron's p	**0.67**
Search for randomization list…	**Unchecked**
Number of Groups	**2**
All Group Sizes are Equal	**Checked**
Equal Group Size	**10**
Grp 1 Label	**High**
Grp 2 Label	**Low**
Storage Tab	
Store Randomization List	**Checked**
Store Subject ID's In	**1**
Store Group Assignments In	**2**

Output

Click the Run button to perform the calculations and generate the following output. Your output will be different because of random assignment.

Results

Summary

Randomization Algorithm	Efron's Biased Coin (p = 0.67)	
Number of Groups	2	
Total Sample Size	20	
Group Sample Sizes	Actual	Target
-- High	9	10
-- Low	11	10

Randomization List

Subject ID	Group Assignment	Largest % Deviation from Target	Cumulative Sample Size (High, Low)
1	High	5.0%	(1, 0)
2	High	10.0%	(2, 0)
3	Low	5.0%	(2, 1)
4	High	10.0%	(3, 1)
5	Low	5.0%	(3, 2)
6	Low	0.0%	(3, 3)
7	Low	5.0%	(3, 4)
8	High	0.0%	(4, 4)
9	Low	5.0%	(4, 5)
10	High	0.0%	(5, 5)
11	Low	5.0%	(5, 6)
12	High	0.0%	(6, 6)
13	Low	5.0%	(6, 7)
14	High	0.0%	(7, 7)
15	Low	5.0%	(7, 8)
16	Low	10.0%	(7, 9)
17	High	5.0%	(8, 9)
18	High	0.0%	(9, 9)
19	Low	5.0%	(9, 10)
20	Low	10.0%	(9, 11)

In this case, no search was conducted. The actual and target group sample sizes turned out to be slightly different. Go to the data window to view the results stored there.