

Create Random Groups in Excel (With Example)

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Create Random Groups in Excel (With Example)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=92607>

It is frequently necessary in data analysis and resource management to distribute a list of items or individuals into smaller, randomized cohorts. Generating random groups within a spreadsheet program like Excel allows for unbiased assignment, which is critical for tasks ranging from classroom organization to experimental design or, as we will demonstrate here, forming sports teams. This tutorial provides a comprehensive, step-by-step methodology for executing this task efficiently using a combination of powerful built-in functions.

Consider a practical scenario: you have a roster of twelve basketball players who must be divided equally and randomly into three distinct teams. Achieving this randomization manually can be tedious and prone to human bias. However, utilizing specific functions in Excel enables the creation of a dynamic grouping system that ensures fairness and reproducibility. The following example illustrates how the players are seamlessly allocated to teams 1, 2, or 3 based purely on random values generated by the software.

The image below visualizes the desired outcome, where each player listed in Column A is assigned a corresponding random team number in Column C. This assignment relies on an intermediary step--generating a unique random numerical value for each individual, which dictates their eventual rank and group placement.

C2 $\times$ $\checkmark$ f_x =ROUNDUP(RANK(B2,\$B\$2:\$B\$13)/4,0)

	A	B	C	D	E	F	G
1	Player	Rank	Team				
2	Andy	0.392495	2				
3	Bob	0.214684	2				
4	Chad	0.839582	1				
5	Doug	0.085107	3				
6	Eric	0.459054	1				
7	Frank	0.214952	2				
8	Greg	0.026702	3				
9	Henry	0.142699	3				
10	Isaac	0.526874	1				
11	John	0.052288	3				
12	Kendall	0.554933	1				
13	Luke	0.205055	2				
14							
15							

Fortunately, implementing this sophisticated randomization technique is straightforward once the underlying logic is understood. The subsequent steps will guide you through the initial data entry,

the generation of random values, and the final application of the grouping formula, providing a robust solution for any scenario requiring randomized assignment.

Step 1: Preparation and Data Entry

The initial phase of creating random groups involves meticulously structuring the raw data within the Excel spreadsheet. Accuracy in this step is paramount, as all subsequent calculations depend on the integrity and organization of the source list. Begin by entering all the items or names you wish to group into a single, contiguous column, typically starting in cell A2, leaving A1 for a descriptive header like "Player Name."

For our running example, we must enter the names of the twelve basketball players who are to be randomly distributed across three teams. It is essential to ensure that the list is complete and correct before proceeding, as adding or removing entries later will require adjusting the formula ranges used in subsequent steps. This preparation ensures a clean foundation for the randomization process.

	A	B	C	D	E
1	Player				
2	Andy				
3	Bob				
4	Chad				
5	Doug				
6	Eric				
7	Frank				
8	Greg				
9	Henry				
10	Isaac				
11	John				
12	Kendall				
13	Luke				
14					
15					
16					

Once the names are listed, designate an adjacent column, such as Column B, to store the generated random values, and Column C for the final group assignment. Clearly labeling these columns--perhaps "Random Value" and "Assigned Team"--enhances readability and makes the

spreadsheet easier to audit and maintain. This organization is key to understanding the mechanism that drives the final grouping.

Step 2: Generating the Core Random Values

The fundamental mechanism for achieving true random assignment in Excel relies on generating a unique, uniformly distributed random number for every entry in the list. This numerical assignment is executed using the built-in RAND() function, which is designed to return a volatile decimal value greater than or equal to 0 and less than 1. This function is ideal because it requires no arguments and recalculates whenever the spreadsheet is changed, ensuring fresh randomization upon every modification.

To implement this, navigate to the first empty cell in the designated random value column--in our case, cell **B2**--and input the function exactly as shown below. The simplicity of this function belies its critical importance; it is the source of the necessary variation required to break any existing order in the player list.

=RAND()

After entering the formula in cell **B2**, apply it to the remaining players. This is efficiently achieved by clicking on the fill handle (the small square at the bottom-right corner of the selected cell) and dragging it down until the function is copied for every player in the list, concluding at cell B13. This action instantaneously populates Column B with twelve distinct random decimal numbers.

B2		=RAND()				
	A	B	C	D	E	
1	Player	Rank				
2	Andy	0.913388				
3	Bob	0.606517				
4	Chad	0.841348				
5	Doug	0.367257				
6	Eric	0.056275				
7	Frank	0.773275				
8	Greg	0.852885				
9	Henry	0.17997				
10	Isaac	0.45566				
11	John	0.756909				
12	Kendall	0.858313				
13	Luke	0.215071				
14						
15						
16						

As a result of this step, every player now possesses an associated random numerical value between 0 and 1. These values serve as the randomization key; the grouping formula in the next step will use the numerical rank of these random values--not the alphabetical order of the names--to determine team assignment, thus ensuring an equitable distribution.

Step 3: Calculating Numerical Rank using RANK

The core challenge in random grouping is not just generating random numbers, but converting those continuous decimal values into discrete, manageable group identifiers (e.g., Team 1, Team 2, Team 3). This transformation requires determining the relative position of each player's random number within the entire dataset. This is achieved through the utilization of the RANK() function.

The RANK() function identifies the position of a specific number within a list of numbers. When applied to the random values generated in Column B, it effectively assigns a unique integer rank from 1 to 12 (since there are twelve players) based on the magnitude of the random number. For example, the player with the highest random value will receive rank 1 (if sorted descendingly), while the player with the lowest value receives rank 12. This step mathematically orders the previously unordered list.

To perform this operation, we integrate RANK() into the overall grouping formula. The syntax

requires specifying the number to rank (the individual player's random value) and the reference array (the entire column of random values). Crucially, the reference array must be absolute (using dollar signs, e.g., **\$B\$2:\$B\$13**) to prevent the range from shifting when the formula is copied down, ensuring all players are ranked against the same set of random numbers.

Step 4: Implementing the Final Group Allocation Formula

The combined formula, which we will now enter into cell **C2**, utilizes the calculated rank and a specific division factor to produce the final group assignment. This complex formula converts the sequential ranks (1 through 12) into a repeating series of group numbers (1, 2, 3, 1, 2, 3, etc.), effectively sorting the players into equal, randomized teams.

The complete formula to assign each player to one of three random groups is:

=ROUNDUP(RANK(B2,\$B\$2:\$B\$13)/4,0)

Once this formula is entered into cell **C2**, it must be copied down to the remaining cells in Column C. This assignment process is immediate, resulting in the desired outcome where twelve players are systematically sorted into three teams of four players each.

		C2 : =ROUNDUP(RANK(B2,\$B\$2:\$B\$13)/4,0)					
	A	B	C	D	E	F	G
1	Player	Rank	Team				
2	Andy	0.392495	2				
3	Bob	0.214684	2				
4	Chad	0.839582	1				
5	Doug	0.085107	3				
6	Eric	0.459054	1				
7	Frank	0.214952	2				
8	Greg	0.026702	3				
9	Henry	0.142699	3				
10	Isaac	0.526874	1				
11	John	0.052288	3				
12	Kendall	0.554933	1				
13	Luke	0.205055	2				
14							
15							

Upon applying the formula, Column C now displays the final group assignments. Because the

rankings are determined by the volatile `RAND()` function in Column B, the team assignments will change every time the sheet recalculates, demonstrating true randomization.

For example, based on the specific random values generated in the image above, the results show a clear distribution:

Andy has been assigned to Team 2.

Bob has been assigned to Team 2.

Chad has been assigned to Team 1.

This process continues down the list, ensuring that the 12 players are divided into approximately equal groups, providing an unbiased starting point for any team activity or experimental setup.

Step 5: Deconstructing the Grouping Logic

To fully understand the effectiveness of this technique, it is vital to dissect the role of the final mathematical operators: the division factor and the `ROUNDUP()` function. The purpose of this combination is to segment the continuous range of ranks (1 through 12) into discrete blocks corresponding to the desired number of teams.

In the formula `=ROUNDUP(RANK(...)/4, 0)`, the value **4** represents the crucial division factor. This number specifies the maximum number of items (players) that should be assigned to any single group. Since we have 12 players and desire 3 teams, 12 divided by 3 equals 4 players per team. Therefore, dividing the rank by 4 creates three distinct numerical ranges:

Ranks 1, 2, 3, and 4, when divided by 4, result in a number less than or equal to 1.

Ranks 5, 6, 7, and 8, when divided by 4, result in a number greater than 1 but less than or equal to 2.

Ranks 9, 10, 11, and 12, when divided by 4, result in a number greater than 2 but less than or equal to 3.

The outer wrapper, `ROUNDUP()`, is essential because it forces these resulting fractional numbers up to the nearest integer. Specifically, `ROUNDUP(..., 0)` ensures that any value between 0 and 1.0 (exclusive of 0, due to the rank starting at 1) becomes 1, any value between 1.01 and 2.0 becomes 2, and so forth. This mechanism successfully translates the ranked order into defined group identifiers (1, 2, and 3) based on the size constraint (4 players) set by the divisor.

Step 6: Customizing Group Sizes and Ratios

A significant advantage of this formulaic approach is its flexibility. The number of groups created and the size of those groups are entirely controlled by the division factor placed immediately after

the `RANK()` function output. If your requirements change--for instance, if you need fewer, larger teams--you simply adjust this numerical input.

For our dataset of 12 players, suppose the requirement shifts from three teams of four players to two teams of six players. In this scenario, the division factor must be adjusted to 6. This is calculated by dividing the total number of players (12) by the desired number of teams (2), yielding 6. By changing the divisor from 4 to 6, the formula adjusts the cutoff points for group assignment, ensuring an equal split between the two groups.

To implement this modification, the formula in cell **C2** is updated as follows:

=ROUNDUP(RANK(B2,\$B\$2:\$B\$13)/6,0)

After making this change, type the formula into cell **C2** and then utilize the fill handle to drag it down through the rest of Column C. The groups will instantly recalculate based on the new divisor, reflecting the assignment of six players to Team 1 and six players to Team 2.

	A	B	C	D	E	F	G
1	Player	Rank	Team				
2	Andy	0.427137	1				
3	Bob	0.311743	2				
4	Chad	0.396126	1				
5	Doug	0.142605	2				
6	Eric	0.324845	2				
7	Frank	0.363186	1				
8	Greg	0.117371	2				
9	Henry	0.23168	2				
10	Isaac	0.800866	1				
11	John	0.528999	1				
12	Kendall	0.466182	1				
13	Luke	0.358589	2				
14							
15							
16							
17							

As clearly illustrated in the image above, the resulting distribution now adheres to the specified constraints: six players are assigned to Team 1, and the remaining six players are assigned to Team 2. This flexibility allows for dynamic adjustment of grouping criteria without altering the core structure of the randomization technique. Always remember that the value used after the division

symbol directly dictates the target number of members in each generated group.

ARABPSYCHOLOGY.COM