

# How to Randomize a List in Google Sheets: A Step-by-Step Guide

Authored by  
**stats writer**

December 3, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Randomize a List in Google Sheets: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=104112>

The ability to introduce controlled randomization into your datasets is a critical feature for anyone managing large tables or conducting statistical analysis. Google Sheets provides intuitive tools to achieve this randomization efficiently. Unlike dedicated formula functions like `RAND()` or `SORT()`, the dedicated **Randomize Range** tool allows users to quickly and permanently shuffle the order of selected cells or rows with just a few clicks. This built-in feature is invaluable when generating truly random samples, creating unbiased groups for experimental designs, or simply shuffling a list of names for fairness in team selection.

Understanding how to properly utilize the **Randomize Range** feature is essential, especially when dealing with associated data columns where maintaining the integrity of paired values is paramount. This guide will explore the specific mechanisms of the user interface (UI) randomization tool, detail its application across single and multiple ranges, and provide essential best practices to ensure your data remains accurate throughout the shuffling process. We will begin by looking at the basic requirements for initiating a randomized shuffle within your spreadsheet environment.

This randomization process is highly useful for various professional and personal applications, including generating lottery numbers, simulating probabilistic outcomes, or anonymizing the order of entries before presentation. Whether you are a beginner or an advanced spreadsheet user, mastering this tool will significantly enhance your Google Sheets proficiency.

In practical spreadsheet management, the need to randomly reorder items within one or more lists arises frequently. This is often necessary to eliminate biases introduced by the initial ordering or to prepare data for sampling techniques.

Fortunately, performing a thorough shuffle in Google Sheets is remarkably straightforward. The following detailed examples illustrate how to implement this process effectively across various data structures, ensuring optimal results regardless of whether you are handling a single column or multiple linked columns of information.

## The Quick UI Tool - Randomize Range Feature

The most accessible way to achieve permanent randomization in Google Sheets is through the dedicated UI tool, **Randomize Range**. This feature is preferred when you need a static, irreversible reordering of your selected cells without relying on complex array formulas. Crucially, the randomization is executed instantly and overwrites the existing order, making the operation fast but permanent. Due to its permanent nature, it is always recommended to duplicate the sheet or column prior to execution if you need to preserve the original sequence.

The **Randomize Range** tool utilizes an internal pseudo-random algorithm to assign a unique, random sort key to each selected row, effectively scrambling the list. This method is highly reliable

for generating an unbiased sequence. It is housed under the Data menu, making it easily accessible even for users who are unfamiliar with spreadsheet functions. The key advantage of this method is its simplicity; it requires no formula construction or understanding of volatile functions like `RAND()`.

Before initiating the process, carefully define your range. Selection accuracy is vital, especially concerning headers. If a header row is accidentally included in the selected range, it will be treated as standard data and will also be randomized, potentially placing the header label somewhere in the middle of your list. We will now explore two primary scenarios: randomizing a standalone list and randomizing two or more linked lists while maintaining their row relationships.

### Example 1: Randomize One List in Google Sheets

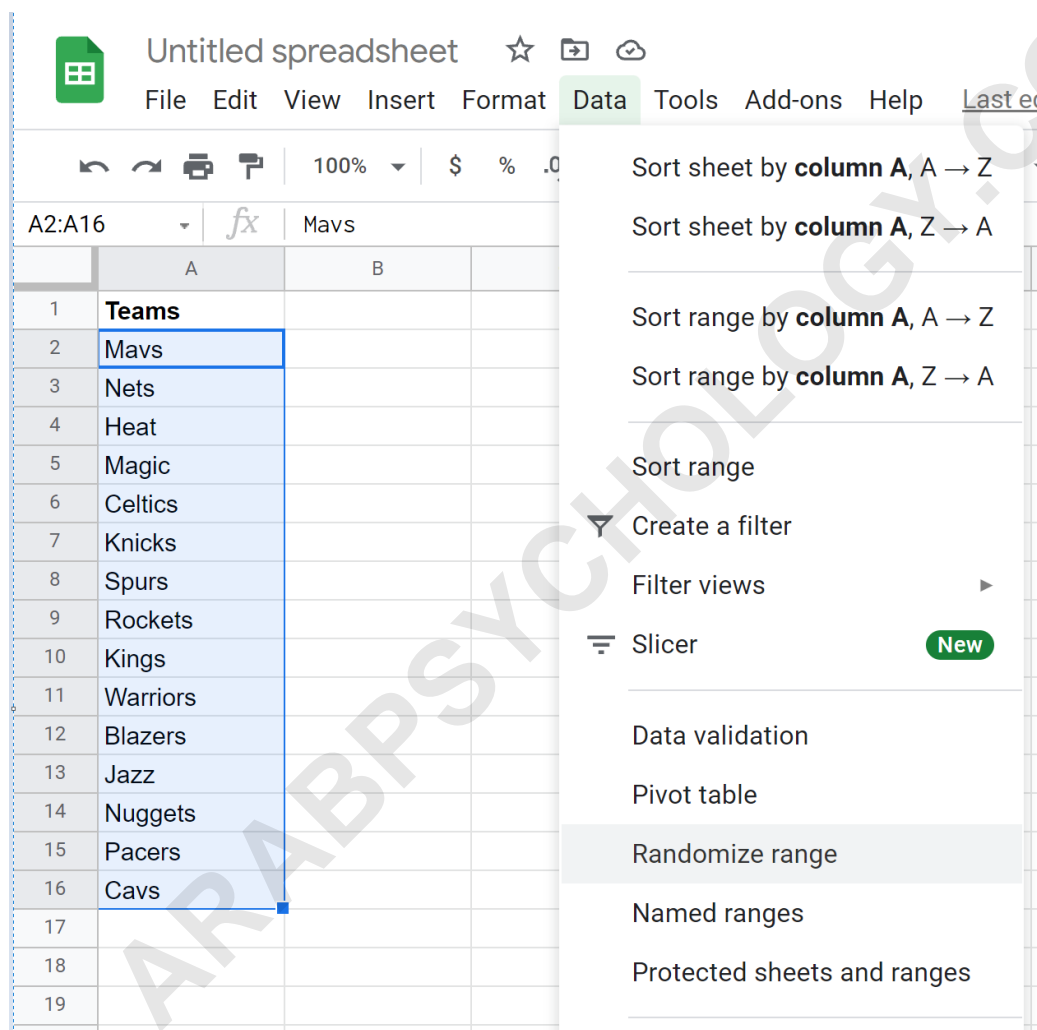
Consider a common scenario where you possess a single list that needs to be randomly shuffled, such as a list of candidates, survey IDs, or, as demonstrated here, a list of team names. Suppose we have the following list of basketball team names in Google Sheets, located in Column A:

	A	B	C	D	
1	<b>Teams</b>				
2	Mavs				
3	Nets				
4	Heat				
5	Magic				
6	Celtics				
7	Knicks				
8	Spurs				
9	Rockets				
10	Kings				
11	Warriors				
12	Blazers				
13	Jazz				
14	Nuggets				
15	Pacers				
16	Cavs				
17					
18					
19					
20					
21					
22					

To properly execute the randomization process for this standalone list, the first critical step is

accurate selection. You must highlight only the actual data intended for shuffling. Ensure that you intentionally exclude the header cell ("Teams") from your highlighted range to prevent it from being shuffled into the dataset.

Once the data range (e.g., A2:A11 in this example) is highlighted, navigate to the main menu bar. Click the **Data** tab, which contains all sheet manipulation tools. Within the dropdown menu, locate and click the specific tool labeled **Randomize Range**. This action triggers the randomization algorithm on the selected cells:



Upon completion, the tool will automatically reorder the contents of the cells within your defined range, achieving a newly randomized sequence of the team names. This transformation is immediate and permanent, resulting in a completely scrambled list, similar to the one shown below:

	A	B	C	D	E
1	<b>Teams</b>				
2	Magic				
3	Pacers				
4	Warriors				
5	Kings				
6	Rockets				
7	Knicks				
8	Mavs				
9	Cavs				
10	Heat				
11	Celtics				
12	Nuggets				
13	Spurs				
14	Blazers				
15	Jazz				
16	Nets				
17					
18					
19					
20					
21					
22					

**Important Note on Selection Integrity:** Always verify your selection before clicking **Randomize Range**. As previously emphasized, if you accidentally select the column header (A1, labeled "Teams"), this header value will be treated as another entry and will also be randomized throughout the list. For single-column randomization, selecting only the data rows ensures the structure of your table remains intact.

## Example 2: Randomize Multiple Lists in Google Sheets

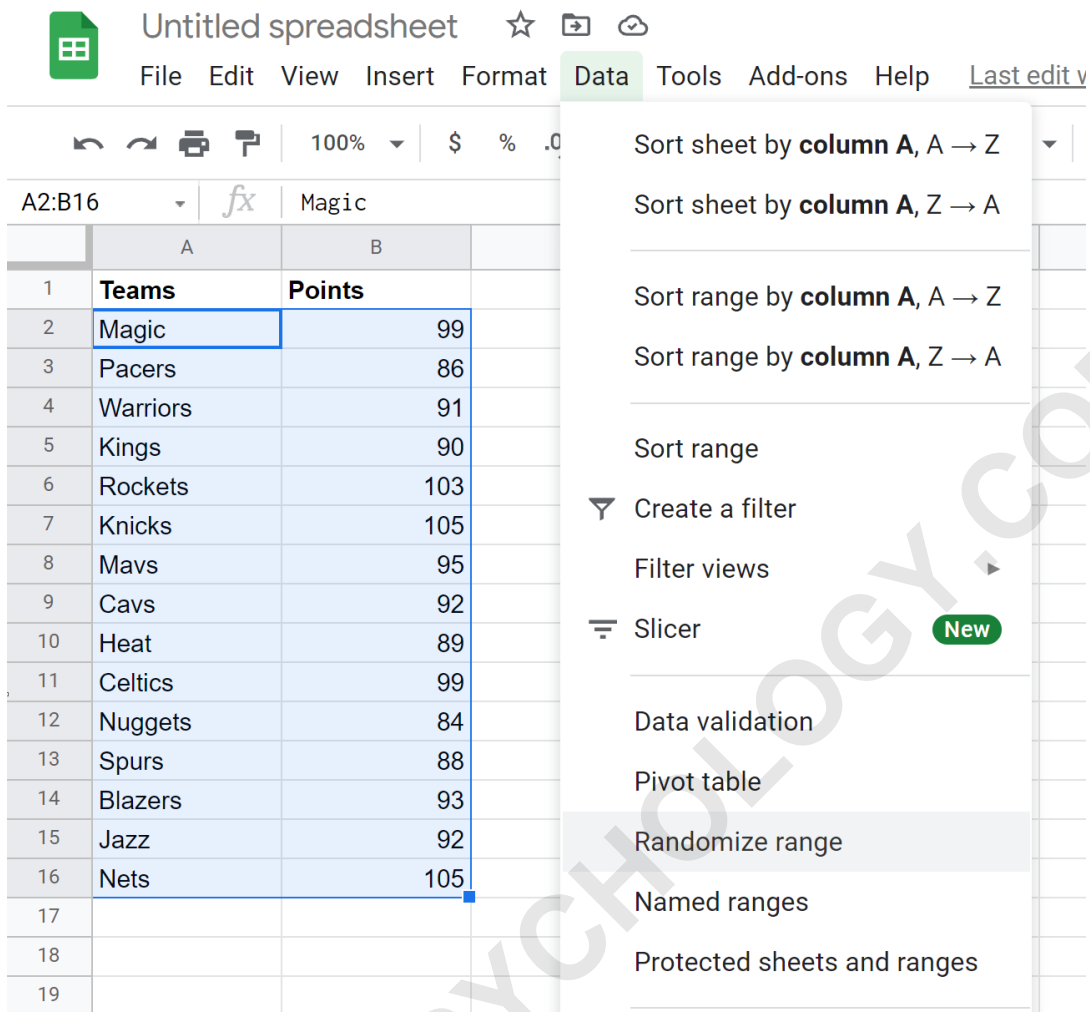
In many real-world applications, data is organized across multiple columns, where each row represents a complete record. For instance, you might have a list of basketball team names paired with their corresponding scores, identification numbers, or player rosters. If you need to shuffle this dataset, it is crucial that the link between the team name and its score is preserved. Shuffling one column independently of the other would corrupt the data integrity.

Suppose we have the following list, containing basketball team names in Column A and their associated points scored in Column B:

	A	B	C	D	E
1	<b>Teams</b>	<b>Points</b>			
2	Magic	99			
3	Pacers	86			
4	Warriors	91			
5	Kings	90			
6	Rockets	103			
7	Knicks	105			
8	Mavs	95			
9	Cavs	92			
10	Heat	89			
11	Celtics	99			
12	Nuggets	84			
13	Spurs	88			
14	Blazers	93			
15	Jazz	92			
16	Nets	105			
17					
18					
19					
20					

To correctly randomize this list while ensuring the teams remain matched with the correct number of points, you must treat the adjacent columns as a single contiguous range. First, highlight the entire block of data, spanning both Column A and Column B (e.g., A2:B11). By selecting all relevant columns, you instruct Google Sheets to treat each row as a single unit during the shuffling process.

With the combined range selected, proceed as before: click the **Data** tab and then click **Randomize Range**. It is essential to understand that the randomization algorithm will apply the same random movement to all cells within the selected row simultaneously, ensuring that 'Lakers' (A cell) moves with '115' (B cell) to its new position:



Untitled spreadsheet

File Edit View Insert Format Data Tools Add-ons Help Last edit v

100% \$ % .0

A2:B16 fx Magic

	A	B
1	<b>Teams</b>	<b>Points</b>
2	Magic	99
3	Pacers	86
4	Warriors	91
5	Kings	90
6	Rockets	103
7	Knicks	105
8	Mavs	95
9	Cavs	92
10	Heat	89
11	Celtics	99
12	Nuggets	84
13	Spurs	88
14	Blazers	93
15	Jazz	92
16	Nets	105
17		
18		
19		

- Sort sheet by **column A**, A → Z
- Sort sheet by **column A**, Z → A
- Sort range by **column A**, A → Z
- Sort range by **column A**, Z → A
- Sort range
- ▼ Create a filter
- Filter views
- ☰ Slicer **New**
- Data validation
- Pivot table
- Randomize range**
- Named ranges
- Protected sheets and ranges

The resulting sheet will show the randomized sequence of teams, but crucially, the structural integrity of the data is preserved. Each team entry remains flawlessly paired with its original point total, proving that the randomization was applied row-wise across the selected columns:

	A	B	C	D
1	<b>Teams</b>	<b>Points</b>		
2	Rockets	103		
3	Celtics	99		
4	Heat	89		
5	Kings	90		
6	Spurs	88		
7	Pacers	86		
8	Magic	99		
9	Knicks	105		
10	Nets	105		
11	Nuggets	84		
12	Cavs	92		
13	Jazz	92		
14	Warriors	91		
15	Blazers	93		
16	Mavs	95		
17				
18				
19				
20				

### Method 3: Dynamic Randomization using the Formula Approach

While the **Randomize Range** UI tool is quick and effective for a permanent shuffle, it does not offer dynamic results. If your source data changes frequently, or if you need a non-destructive method that preserves the original list while generating a randomized output list that updates automatically, the formula approach is necessary. This method leverages two primary functions: `RAND()` and `SORT()`.

The core concept involves generating a temporary, adjacent column filled with random decimal numbers between 0 and 1 using the volatile function `RAND()`. Since `RAND()` generates a new number every time the spreadsheet recalculates (or is opened), these random values serve as unique, constantly changing keys. We then use the `SORT()` function (or `QUERY()`) to reorder the original list based on these ever-changing random keys.

To implement this, first insert a new column (e.g., Column B) next to your original list (Column A). In the first cell of the new column (B2), input the formula `=RAND()` and drag it down to cover the same number of rows as your list. You will then use the `SORT()` function in a separate output area (e.g., Column D) to pull the randomized list dynamically. The formula would look something like

`=SORT(A2:A11, B2:B11, TRUE)`, where the `TRUE` parameter indicates ascending order based on the random numbers in Column B. Because the `RAND()` values constantly change, the output list in Column D will automatically shuffle every time the sheet recalculates, providing true dynamic randomization.

## Advanced Technique: Using RANDARRAY for Efficiency

For modern spreadsheet users utilizing the most recent features of Google Sheets, the `RANDARRAY()` function offers a more concise and efficient way to generate the necessary random sort keys, especially useful when combined with the `SORT()` function within a single formula. Unlike `RAND()`, which must be dragged down manually, `RANDARRAY()` can instantly generate an entire array of random numbers based on specified dimensions.

If your original list is in A2:A11 (10 rows), you can generate the required 10 random sort keys using `RANDARRAY(10, 1)`. This powerful array function allows for an elegant, single-cell formula solution that avoids the need for creating a helper column entirely. The entire dynamic randomization can be executed in one step, using the formula structure `=SORT(A2:A11, RANDARRAY(ROWS(A2:A11), 1), TRUE)`.

This single-cell approach significantly cleans up the spreadsheet and reduces the risk of accidental data modification, as the random keys are generated transiently within the formula execution itself. The `ROWS()` function ensures that the number of random keys generated matches the exact size of your input range, making the formula robust and scalable. Whenever the input data in A2:A11 is updated or the sheet is refreshed, this formula will instantly produce a new, randomized order.

## Comparing Methods: UI Tool vs. Formula

Choosing between the permanent UI tool (Randomize Range) and the dynamic formula approach (`SORT/RAND`) depends entirely on the specific requirements of the project and the desired behavior of the output data. Both methods are valid means of achieving randomization, but their practical implications differ significantly regarding performance, volatility, and ease of use.

The **Randomize Range** tool is superior for scenarios where speed and a non-volatile result are paramount. Because the result is permanent, it eliminates constant recalculations, which can sometimes slow down very large spreadsheets. It is an excellent choice for one-off tasks like assigning teams, shuffling presentation order, or generating a static sample list. However, it lacks automation; every time a shuffle is required, the user must manually access the Data menu and execute the command again, and the original data is overwritten.

Conversely, the **Formula Approach**, particularly using `SORT` and `RANDARRAY`, provides dynamic, automated output. This is ideal for simulations, statistical modeling where continuous re-sampling

is needed, or dashboards that require a constantly changing display order. The main drawback is the volatility introduced by the `RAND()` or `RANDARRAY()` functions. Every user interaction or data change causes a re-sort, which can sometimes be distracting or taxing on resources in complex sheets. Furthermore, it requires a slightly higher level of technical knowledge to implement correctly.

## Best Practices for Effective Data Randomization

Regardless of the method chosen, adhering to certain best practices ensures that your randomization is statistically valid and that your underlying data remains secure and intact. Proper preparation minimizes errors and maximizes the utility of the shuffling process for research or administrative tasks.

**Back Up Your Data:** Always duplicate the sheet or copy the original range into a new location before using the **Randomize Range** tool, as this operation is destructive and cannot be undone via a simple refresh or formula change.

**Verify Range Selection for Paired Data:** When randomizing multiple columns (like in Example 2), confirm that you have selected all relevant columns that constitute a complete record. Failure to select the entire row range will result in corrupted data pairings.

**Exclude Headers Accurately:** Ensure that the header row is excluded from the randomization selection for both the UI tool and the formula inputs, unless the header row itself is intended to be part of the randomized dataset.

**Understand Volatility:** If using the formula method (`RAND()`), be aware that the sheet will constantly update. If you need a specific randomized order to be static after generating it dynamically, you must copy the output column and paste it as static values (Paste Special > Paste values only).

## Conclusion: Mastering Random Order Generation

Both the quick UI solution and the robust formula-based methods offer powerful avenues for achieving controlled randomization within your spreadsheet workflows. The **Randomize Range** feature provides an immediate, user-friendly way to permanently shuffle lists, perfectly suited for static, operational requirements. Conversely, the combination of `SORT()` and `RANDARRAY()` delivers a dynamic, automated solution for ongoing analysis and simulations.

By understanding the nuances of these two approaches--the speed and finality of the UI tool versus the adaptability and automation of the formula algorithm--you can select the appropriate technique for any given task in Google Sheets. Mastery of these randomization techniques is fundamental for anyone looking to perform unbiased sampling or introduce fair variability into their spreadsheet management.

The following examples show how to perform other common operations in Google Sheets:

ARABPSYCHOLOGY.COM