

Learning to Sort and Synchronize Two Columns in Google Sheets

Authored by
Mohammed loot

November 11, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Sort and Synchronize Two Columns in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16933>

In the realm of advanced [Google Sheets](#) data management, users frequently encounter the challenge of synchronizing the order of entries across two separate columns or lists. This essential technique, often referred to as a **synchronized sort** or **matched sort**, is vital for maintaining the established relationships between corresponding data points. For instance, if you have a list of product names in Column A and associated inventory counts in Column B, sorting Column A alphabetically without linking the two columns would immediately corrupt your dataset. This tutorial provides a meticulous, step-by-step methodology detailing how to achieve perfect positional alignment between two lists within Google Sheets, ensuring that the target column precisely mirrors the sequence of a designated reference column.

The standard sorting mechanisms available in spreadsheet software typically operate on a single column or treat multiple columns alphabetically, which fails when the goal is to align List A based on the specific, non-alphabetical order of List C. Our solution circumvents this limitation by introducing an intermediary step: the creation of a unique numerical index. By leveraging powerful lookup functions, specifically the [MATCH Function](#), we translate the desired order of the reference column (List C) into a numerical priority key. This key then dictates the final arrangement of the primary column (List A), guaranteeing absolute **data integrity** throughout the process.

This method is highly effective for reconciling related datasets, such as mapping employee names to unique security IDs or matching statistical results to their corresponding trial numbers. The key insight is that we are not sorting the text itself, but rather sorting the numerical values assigned to that text, thereby forcing the text to adhere to a customized sequence. This preparatory work is crucial for any rigorous data analysis or comparison where the sequential relationship between two lists must be preserved or enforced.

Setting the Stage: Structuring the Initial Data

Before initiating the specialized sorting routine, it is necessary to properly structure the base data set. For illustrative purposes, we will use two distinct lists of basketball team names. It is critical to recognize that while these lists are fundamentally related, they currently exist in divergent, unsynchronized orders. Our specific objective is to rearrange the first list (Column A) so that its sequence perfectly aligns with the exact sequence found in the second list (Column C). This scenario mirrors common real-world requirements where two data entry points or merged data files need to be harmonized according to a specific, master reference order.

To successfully implement this technique, you must enter the data values into your [spreadsheet](#), ensuring that the two lists occupy non-adjacent columns--in our example, Columns A and C. This deliberate separation allows us sufficient room to insert the necessary calculation column (Column B), which will serve as our temporary **sorting index** or key. Column A is designated as **Team List 1** (the list we intend to sort and modify), and Column C is established as **Team List 2** (the fixed

reference column that provides the desired final order). The successful execution of the entire operation hinges on the proper setup of these three columns.

The specific team names utilized here are examples only, but the underlying methodology is universally applicable to any type of data, whether textual, numerical, or date-based, that requires synchronized reordering. Below is the initial configuration of the data, spanning the [cell range](#) A2 through C11. Note the current lack of synchronization: the team "Lakers" in cell A2 needs to move to align with the position of "Lakers" in C5 if the orders are to be matched:

	A	B	C	D
1	Team List 1		Team List 2	
2	Mavs		Kings	
3	Warriors		Nets	
4	Hawks		Suns	
5	Kings		Mavs	
6	Suns		Thunder	
7	Grizzlies		Hawks	
8	Nets		Heat	
9	Thunder		Magic	
10	Magic		Grizzlies	
11	Heat		Warriors	
12				
13				
14				
15				

Our preparatory goal is to manipulate the data in Column A until every entry, such as "Lakers," sits directly opposite its counterpart in Column C. This ensures that the sequence of Column A exactly mirrors the sequence of Column C, using a sorting mechanism that is applied only to the pairing of the target column (A) and its adjacent index column (B). This step confirms the data's readiness before we proceed to the critical calculation phase.

Creating the Numerical Sort Key with the MATCH Function

The core innovation of this synchronized sorting technique is the generation of a numerical index that effectively maps the position of every value in Column A relative to its location within the reference list, Column C. This mapping is executed flawlessly using the powerful [MATCH function](#). The primary utility of this function here is not to return the value itself, but rather the **relative row number** where the value from List 1 is found within the specified range of List 2. Once this numerical position is identified for every item, we possess the customized, sequential sorting key

required for the final alignment.

The standard syntax for the **MATCH** function is straightforward: `MATCH(search_key, range, search_type)`. Applied to our scenario, the formula is meticulously crafted to look up the team name found in the current row of Column A (the `search_key`) and determine its precise position within the fixed range established by Column C (the `range`). The final argument, specified as `0`, is paramount because it enforces an **exact match** requirement. This strict setting guarantees that the returned position is perfectly accurate, even if the list in Column C is unsorted, thereby maintaining the highest degree of data accuracy--a vital consideration for preventing misalignment during the subsequent sort operation.

To implement this, enter the following formula into cell **B2**. Pay close attention to the use of **absolute references** (indicated by the dollar signs: `$C$2:$C$11`) applied to the search range. This referencing technique is essential because while the search key (A2, A3, A4, etc.) must dynamically change as the formula is copied down the column, the reference list it is searching within (Column C) must remain fixed and consistent across all calculations. This ensures that every team in Column A is measured against the identical reference sequence in Column C:

=MATCH(A2, \$C\$2:\$C\$11, 0)

After the formula is accurately placed in B2, utilize the fill handle to drag the formula down to cell B11. Column B will instantly populate with numerical values, where each number represents the sequential position of the corresponding Column A team name within the Column C list. For example, if "Lakers" (A2) is the fifth entry in the range C2:C11, cell B2 will display the number 5. This calculated column now functions as our sophisticated index, assigning a numeric rank to every item in Column A based solely on its desired position according to Column C.

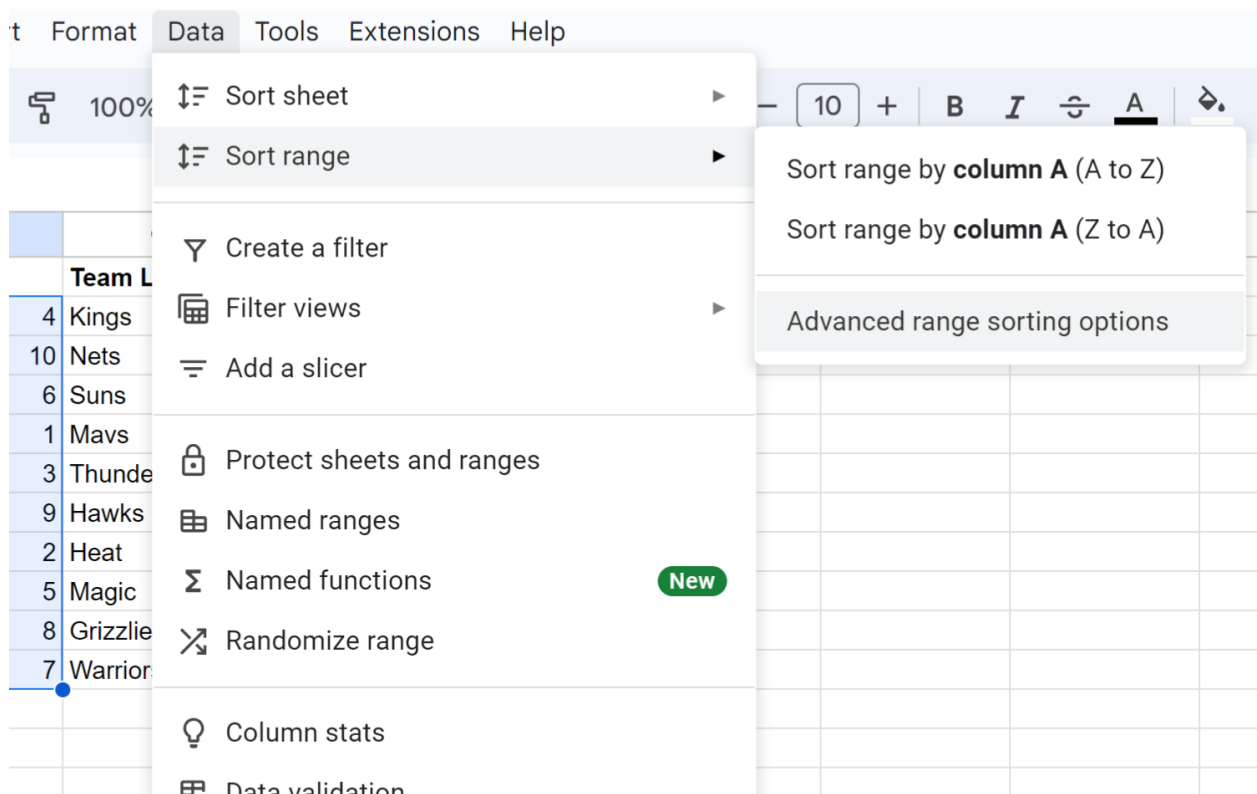
B2 =MATCH(A2, \$C\$2:\$C\$11, 0)

	A	B	C	D
1	Team List 1		Team List 2	
2	Mavs	4	Kings	
3	Warriors	10	Nets	
4	Hawks	6	Suns	
5	Kings	1	Mavs	
6	Suns	3	Thunder	
7	Grizzlies	9	Hawks	
8	Nets	2	Heat	
9	Thunder	5	Magic	
10	Magic	8	Grizzlies	
11	Heat	7	Warriors	
12				
13				
14				
15				

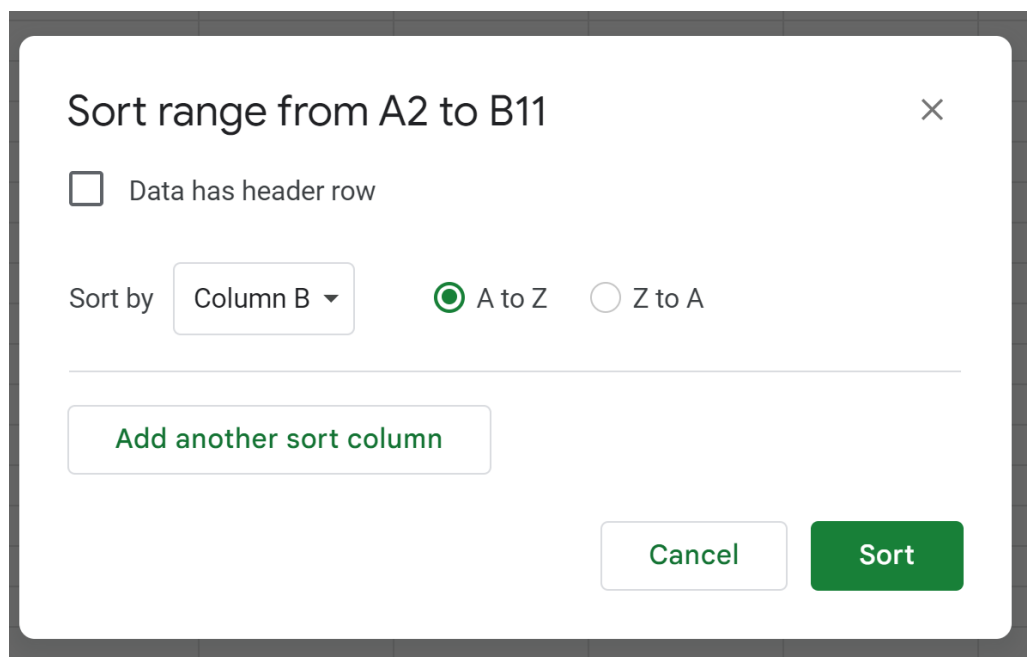
Executing the Advanced Synchronized Sort

With the sorting key successfully established in Column B, the final crucial step involves applying the sort operation. It is imperative to grasp that we are not sorting Column A (the team names) based on its alphabetical content, but rather sorting the selected block of data (Columns A and B) based exclusively on the numerical sequence provided by Column B. This action forces the text values in Column A to move in lockstep with their associated index numbers in Column B, culminating in the precise alignment dictated by the reference column, C.

To begin the process, accurately highlight the entire data set that encompasses the list to be sorted and its corresponding index key. Select the [cell range A2:B11](#). A critical point of execution is to deliberately exclude Column C from this selection, as Column C must remain static to preserve the reference order we are attempting to mirror. Including Column C would cause all three columns to sort based on the first column (A), thereby destroying the positional integrity established by the index. Once the correct range is highlighted, navigate to the top menu ribbon and click the **Data** tab. From the subsequent dropdown menu, choose **Sort range**, followed by the selection of **Advanced range sorting options**.



The appearance of the advanced sorting dialog box grants precise control over the sorting criteria. Ensure that the option "Data has header row" remains unchecked, as our selection starts with data, not labels. Within the sorting criteria configuration, you must explicitly specify that the data should **Sort by Column B**. Since Column B contains index numbers (1, 2, 3, 4, etc.) representing the desired order, sorting this column in **ascending order (A to Z)** will naturally reorganize the accompanying text values in Column A according to the sequential positional hierarchy derived from Column C. Once configured, confirm your settings and click the **Sort** button.



Following the execution of the sort, a visible transformation occurs within the selected range (A2:B11). The index numbers in Column B are now perfectly sorted numerically (1, 2, 3, 4, ...), and the team names in Column A have shifted their positions to correspond exactly with these ranks. This result demonstrates the successful achievement of the synchronized sort: the team names in Column A now precisely match the sequence of their counterparts in Column C. This permanent rearrangement confirms that the data is now fully harmonized.

	A	B	C	D	
1	Team List 1		Team List 2		
2	Kings	1	Kings		
3	Nets	2	Nets		
4	Suns	3	Suns		
5	Mavs	4	Mavs		
6	Thunder	5	Thunder		
7	Hawks	6	Hawks		
8	Heat	7	Heat		
9	Magic	8	Magic		
10	Grizzlies	9	Grizzlies		
11	Warriors	10	Warriors		
12					
13					
14					

Finalizing the Dataset: Cleanup and Verification

Once the synchronization is complete and you have rigorously verified that the sequence of Column A perfectly aligns with Column C, the temporary index created in Column B has fulfilled its purpose and is no longer required for maintaining the order. Its function was purely procedural--it served as the necessary key to dictate the sorting action. Because the values in Column A have been permanently moved into their desired final positions, the index column can now be safely removed to enhance the spreadsheet's readability and cleanliness.

To finalize the process, simply select the entire contents of Column B (cells B2 through B11) and use the spreadsheet's delete or clear contents function. Alternatively, for a quicker approach, you may right-click the Column B header and select the option to delete the entire column. This action leaves you with the two lists, A and C, now sharing the exact same row order, confirming the successful application of the [MATCH Function](#) and advanced sorting methods.

	A	B	C	D	
1	Team List 1		Team List 2		
2	Kings		Kings		
3	Nets		Nets		
4	Suns		Suns		
5	Mavs		Mavs		
6	Thunder		Thunder		
7	Hawks		Hawks		
8	Heat		Heat		
9	Magic		Magic		
10	Grizzlies		Grizzlies		
11	Warriors		Warriors		
12					
13					
14					
15					

This technique is highly scalable and is a robust methodology for ensuring long-term **data integrity** when comparing or merging two related datasets that may have drifted out of sequence. It provides a powerful, permanent rearrangement solution, offering a clear alternative to more complex dynamic array formulas when a simple, one-time structural alignment is required. You now possess two perfectly matched lists based on the sequential order of the reference data in Column C.

Further Exploration of Advanced Google Sheets Operations

Achieving mastery in [Google Sheets](#) data manipulation requires integrating techniques such as synchronized sorting with essential lookup and array functions. If your goal is to expand your proficiency in complex spreadsheet data management, consider exploring resources that build upon the foundational concepts introduced in this guide. These advanced operations address common challenges faced by data analysts and specialists, often involving dynamic data handling and multi-criteria sorting.

While the method detailed above provides an effective solution for a permanent sort, many scenarios require dynamic results that update automatically whenever the source data changes. For such cases, array formulas like `SORT`, combined with `ARRAYFORMULA` and `QUERY`, offer sophisticated alternatives that generate the sorted list dynamically in a separate location, negating the need for a temporary index column and manual deletion. Understanding these tools allows for the creation of dashboards and reports that react instantly to data updates, significantly enhancing

workflow efficiency.

The following resources offer guidance on other vital operations in [Google Sheets](#), covering sophisticated sorting, lookup logic, and data consolidation:

A comprehensive guide to implementing conditional sorting based on a combination of text values and numerical thresholds.

Using the [VLOOKUP function](#) for efficiently cross-referencing extensive datasets against a standardized master list.

Essential techniques for combining data from multiple external spreadsheets using the IMPORTRANGE function and ensuring consistency across interconnected files.

Advanced methods for data segmentation and filtering using the FILTER function based on multiple or dynamic criteria.