

# How to Sort Two Columns Together in Google Sheets

Authored by  
**stats writer**

January 17, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Sort Two Columns Together in Google Sheets*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126446>

Welcome to this guide on advanced data manipulation within [Google Sheets](#). While standard sorting functions are useful, aligning two columns based on matching values requires a slightly more sophisticated approach than a simple "Sort Range" execution. This technique is vital when working with datasets where the order in one column must be synchronized precisely with the order of another, often disparate, list of items.

The conventional method--selecting both columns and clicking [Sort Range](#)--only allows you to sort based on a single key column, causing the adjacent column's data to move along with it. However, if Column A and Column C contain the same set of items but in a jumbled, mismatched order, we need a mechanism to map the position of items in one list to the position of items in the other. This guide provides a detailed, step-by-step solution utilizing the powerful [MATCH function](#) to create a temporary index for perfect alignment.

## The Challenge of Sorting Mismatched Data in Google Sheets

When working with large, complex datasets, analysts frequently encounter situations where two lists contain identical elements but are structured differently. For example, you might have two separate lists of employee names or product IDs, and you need to ensure that the order of the first list (List A) perfectly mirrors the order of the second list (List C). A simple alphabetical sort will only align the lists if they contain the exact same entries in the exact same quantity, which is often not the case in real-world data preparation.

The key to achieving this specific alignment is not to sort the data directly based on the values themselves, but rather to create a numerical index that represents the target position of each item in the source column (Column A) relative to the final desired arrangement (Column C). By generating this index, we effectively create a temporary, auxiliary column that dictates the sorting logic, ensuring a precise match row by row.

This systematic approach involves leveraging a utility column to house the index values. Once this indexing column is generated, a specific type of sort--an advanced range sort--is applied using the index column as the key. This procedure ensures the primary column (Column A) is reordered to reflect the sequence found in the target column (Column C), providing the necessary synchronization for complex comparisons or lookups.

## Overview of the Solution: Leveraging the MATCH Function

The solution hinges on the strategic use of the [MATCH function](#). The primary role of the MATCH function in this context is to return the relative position of an item within a specified range. If we instruct the MATCH function to look up each item from Column A within the range of Column C, the resulting value will tell us exactly which row number in Column C that item appears on.

This row number, generated by the MATCH function, serves as our ideal sorting index. When we sort Column A based on these index numbers in ascending or descending order, we are essentially telling Google Sheets: "Place the item from Column A in the position corresponding to its appearance in Column C." This mechanism guarantees that after the sort operation is complete, the items in Column A will be perfectly ordered according to the sequence established in Column C.

The process is robust and highly reliable, provided that all items in the list you are sorting (Column A) exist within the reference list (Column C). If an item in Column A is missing from Column C, the MATCH function will return an error (usually #N/A), indicating that the item cannot be aligned. Therefore, it is important to ensure data integrity before initiating this advanced sorting routine.

## Step 1: Preparing Your Datasets for Alignment

Before executing the sorting procedure, you must first establish your two lists. For this example, we will use two lists of basketball team names. Note that both lists contain the same teams, but they are intentionally listed in different, arbitrary sequences. Column A represents our source list (the list we intend to reorder), and Column C represents our target list (the desired order).

Begin by entering the sample data into two separate columns in your Google Sheet. It is recommended to label these columns clearly, such as "Team List 1" (Column A) and "Team List 2" (Column C), to maintain clarity throughout the process. Ensure that your header row is distinct from the data rows, as this simplifies the range selection later.

The creation of a temporary third column, Column B, is crucial. This column will hold the numerical index values generated by the MATCH function. While the data shown below uses team names, this technique applies equally well to any dataset, including numerical identifiers, complex strings, or dates, as long as the values are consistent between the two lists.

First, let's enter the following two lists of basketball team names in separate columns:

	A	B	C	D
1	<b>Team List 1</b>		<b>Team List 2</b>	
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				

## Step 2: Implementing the MATCH Function for Indexing

The next critical phase involves calculating the index for every item in our source list (Team List 1). We need to determine the exact row number in the target list (Team List 2) where each corresponding team name appears. This index will serve as our sorting key.

To achieve this, we will use the MATCH function in cell B2. The syntax of the MATCH function requires three arguments: the search key, the range to search within, and the search type. We must use an exact match search to ensure precision.

Type the following formula into cell **B2**:

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

Let's break down this formula: **A2** is the value we are searching for (the team name from the first list). **\$C\$2:\$C\$11** is the lookup range, representing the entirety of the second list; the use of dollar signs (absolute referencing) is essential here so the range does not shift when the formula is dragged down. Finally, **0** specifies that we are looking for an exact match. Once entered, the formula in B2 will return the relative row number (1 in this case, since "Nets" is the first entry in C2:C11) of the team name "Nets" within the C column range.

After entering the formula into B2, you must apply it to the remaining rows in Column B. Click on cell B2 and drag the formula handle down to the end of your dataset, specifically cell B11. This

action populates Column B with the corresponding row indices for every team in Column A, indicating their target position relative to Column C.

Then click and drag this formula down to each remaining cell in column B:

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

	A	B	C	D
1	<b>Team List 1</b>		<b>Team List 2</b>	
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				

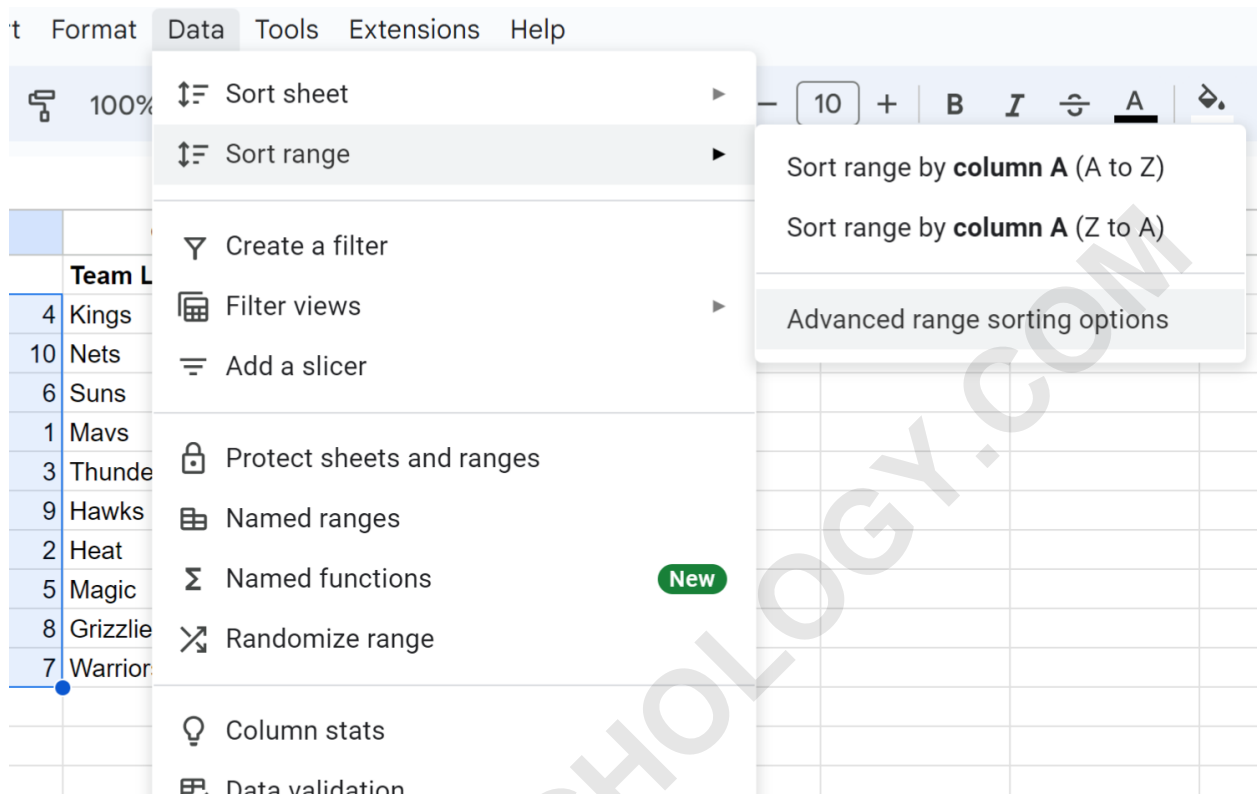
**Note:** The MATCH function is crucial for this procedure because it provides the numerical key needed for a relational sort. For comprehensive documentation on this and other lookup functions in Google Sheets, consult the official Google support resources.

### Step 3: Executing the Advanced Range Sort

With the index column (Column B) fully populated, we are now ready to execute the sort operation. This must be an advanced range sort to ensure that only the source data (Column A) and its corresponding index (Column B) are moved, while the target list (Column C) remains static as our reference.

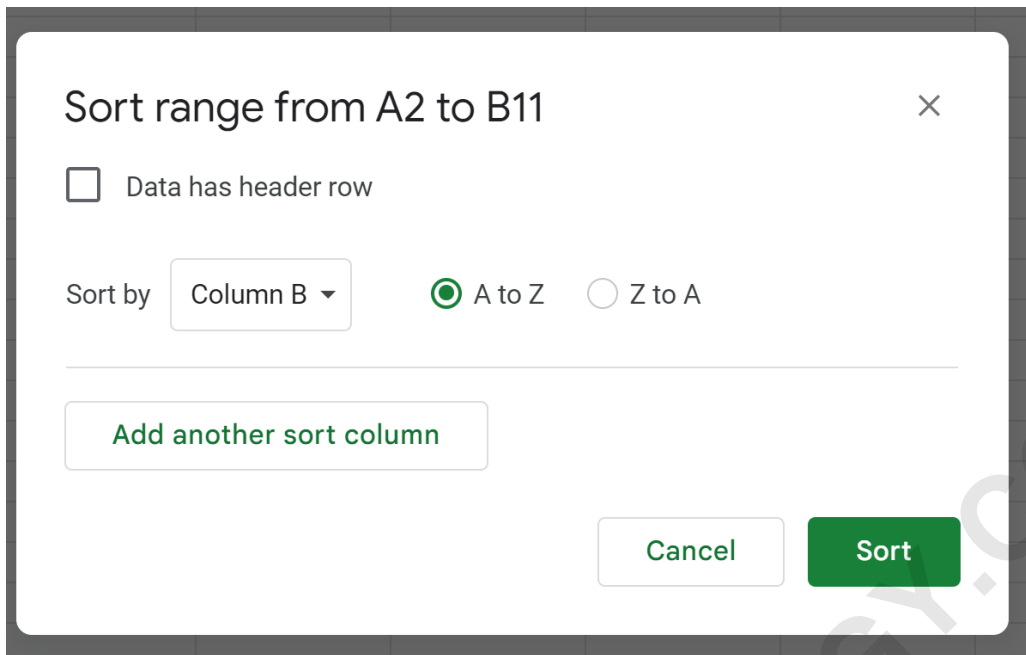
First, highlight the cell range that includes both the source list and the index: **A2:B11**. It is imperative that you include Column B in this selection, as it holds the key for the sort logic. Next, navigate to the main menu ribbon and click the **Data tab**, then select **Sort range**. From the subsequent options, choose **Advanced range sorting options**. This opens a specialized dialogue box that allows for sorting based on multiple or specific columns within a selected range.

Next, highlight the cell range **A2:B11**, then click the **Data tab** along the top ribbon, then click **Sort range**, then click **Advanced range sorting options**:



In the new window that appears, you must specify the key column for the sorting operation. Ensure that you select Column B (the index column) as the key. Since the index numbers represent the ascending row positions (1, 2, 3, etc.) in the target list, the sort order must be **A to Z** (ascending order). If your range included headers, you would also select the option to keep the header row, but since we selected A2:B11, we are only sorting the data itself.

In the new window that appears, choose to **Sort by** Column B from A to Z, then click **Sort**:



### Analyzing the Sorted Output and Verification

Once you click "Sort," the items in Column A will immediately rearrange themselves according to the indices in Column B. Since Column B was sorted numerically from lowest (1) to highest (10), the teams in Column A are now positioned in the exact order dictated by their appearance in the reference list (Column C).

Review the resulting sheet. You will observe that the order of "Team List 1" (Column A) now perfectly matches the order of "Team List 2" (Column C). For instance, if "Warriors" was the first item in Column C, it will now be the first item in the sorted Column A. This confirms the successful execution of the relational sort based on the numerical index derived from the MATCH function.

	A	B	C	D
1	<b>Team List 1</b>		<b>Team List 2</b>	
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				

Since the primary goal was to reorder Column A to match Column C, the temporary index column (Column B) is no longer required for analysis. To maintain a clean and professional spreadsheet environment, you should now proceed to delete the values in Column B, leaving only the newly synchronized data in Column A and the reference data in Column C.

Feel free to delete the values in column B now:

	A	B	C	D
1	<b>Team List 1</b>		<b>Team List 2</b>	
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				

Notice that the team names in each list are now in the exact same order. This method provides a powerful, formula-driven way to align disparate lists without resorting to complex scripting or manual data entry.

## Conclusion: Streamlining Data Analysis with Advanced Sorting

Mastering advanced sorting techniques in [Google Sheets](#) is fundamental for accurate data management and analysis. While the standard **Sort range** command handles basic alphabetical or numerical ordering, integrating the [MATCH function](#) allows for complex relational sorting, enabling users to synchronize data across columns based on value matching rather than inherent alphabetical order.

This technique, involving the creation of a temporary index column and the use of the **Advanced range sorting options** under the **Data tab**, ensures that your source data perfectly mirrors the order of a target reference list. This capability is invaluable for tasks such as preparing merged datasets, validating data consistency between sources, or simply ensuring that two parallel lists are ready for side-by-side comparison. By applying this methodology, you elevate your data manipulation skills in [Google Sheets](#), leading to cleaner, more trustworthy analysis.