

How to Easily Match Data in Two Columns by Sorting in Google Sheets

Authored by
stats writer

February 1, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Easily Match Data in Two Columns by Sorting in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129016>

The imperative of sorting two columns in [Google Sheets](#) to ensure exact alignment is a fundamental requirement for maintaining [data integrity](#) and facilitating robust analysis. This complex sorting scenario goes beyond simple single-column organization; it necessitates arranging the values in two separate columns such that the corresponding entries in each row retain their intended pairing, despite being physically separated. Achieving this alignment relies on a specialized technique: selecting the relevant data area and then employing advanced features like the "[Sort Range](#)" function. Unlike basic sorting, this method utilizes a calculated, common criterion--often a derived numerical index--to organize the data consistently across both columns, whether based on alphabetical labels or numerical magnitude. When both columns are sorted using this identical underlying structure, the data becomes perfectly matched, substantially simplifying subsequent processes such as building aggregated reports, generating visualizations, performing comparative analyses, or effectively managing vast [data sets](#).

This detailed guide outlines an expert methodology, leveraging the power of Google Sheets' built-in functions, specifically the [MATCH function](#), to create a numerical key that dictates the required sorting order. This technique is indispensable when dealing with disorganized or manually entered data that must be reconciled. By following the precise steps detailed below, you can reliably align disparate lists, ensuring that every element in Column A correctly corresponds to its partner in Column B, transforming raw, chaotic input into structured, actionable information ready for high-level processing.

Google Sheets: Advanced Technique to Sort Two Columns for Exact Match

In professional data management, the requirement to align two separate lists or columns based on shared content or a desired sequence is frequent. While [Google Sheets](#) offers powerful sorting capabilities, sorting two columns so that their respective rows match precisely requires a specialized approach that goes beyond standard alphabetical or numerical sorting algorithms.

The following comprehensive, step-by-step example utilizes a conceptual process of creating a temporary index column to enforce the desired matching alignment between two lists, thereby solving the common challenge of disparate data sets that need reconciliation.

Understanding the Necessity for Matched Column Sorting

The need to sort two columns to match typically arises when you have two lists that are conceptually related but physically misaligned. Imagine a scenario where you have a list of items in Column A and their associated weights or values in Column C. If Column A is sorted alphabetically, the values in Column C must follow the exact same positional changes to maintain

the relationship. Without this synchronized movement, the original associations between the data points are destroyed, rendering the resulting analysis meaningless. This is especially true when dealing with critical financial reports or scientific observations where data pairs must remain intact.

This synchronization is generally complex because standard sorting only affects the selected range. If you select Column A and sort it, Column C remains static. Conversely, if you select both A and C, the sort operation typically uses the values in the first selected column (A) as the primary key. If we instead want Column A to conform to the order established by the contents of Column C, a direct sort is impossible unless Column C is physically moved to the left, which is often impractical or detrimental to the overall spreadsheet layout.

Our solution involves defining the desired sequence based on one column (the target sequence column) and then calculating where the elements of the other column (the column to be sorted) appear within that target sequence. This calculation generates a numerical order, or index, which is then applied across both columns simultaneously. By creating this numerical proxy, we enforce a universal sorting rule that respects the relationships between the original data points.

This technique is vital for tasks such as comparing data structures, standardizing input formats across multiple sheets, or preparing data for dashboard integration where consistency of lookup arrays is paramount. It ensures that the integrity of the data pairings is always preserved, regardless of the initial disorder.

Prerequisites and Data Preparation in Google Sheets

Before initiating the sorting procedure, it is essential to ensure your data is correctly structured and prepared within the Google Sheets environment. This process requires two distinct columns of data that hold a one-to-one relationship. In our example, we will use two lists of basketball team names. List 1 represents the data we need to sort, and List 2 represents the desired positional sequence or target order.

First, we must input the two lists into separate, non-adjacent columns. This setup allows for the insertion of a temporary calculation column between them, which is crucial for our indexing strategy. For instance, place the list that needs to be sorted (Team List 1) in Column A and the target list (Team List 2) in Column C. Ensure that the lists occupy the same number of rows to avoid formula errors, even if the values themselves are currently out of order.

Let's begin by entering the values for our example. We will use two lists of basketball team names, demonstrating how to sort Team List 1 (Column A) so that it matches the sequence of Team List 2 (Column C):

	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				

It is strongly recommended to reserve a column immediately adjacent to the column you intend to sort (in this case, Column B, next to Column A) for the calculated index values. This temporary column will hold the results of the [MATCH function](#) and will serve as the sorting key in the subsequent step. Proper preparation of the [data range](#) minimizes potential complications during the advanced sorting process.

Leveraging the MATCH Function for Positional Alignment

The core of this advanced sorting method relies entirely on the powerful capabilities of the Google Sheets [MATCH function](#). This function is typically used to locate the position of a specified value within a one-dimensional array or [range](#). In our context, we repurpose the function to determine the exact row number (or index) where each item from Team List 1 appears within the established order of Team List 2.

By calculating this index for every item in Column A, we essentially assign a numerical rank to each team based on its position in Column C. If "Lakers" is the third entry in Column C, the [MATCH function](#) returns the number 3 for the "Lakers" entry in Column A. This numerical rank is the key to our synchronization, as it mathematically defines the desired position of the item in the final sorted list. When we sort the combined range (A:B) based on this newly generated index column (B), the teams in Column A automatically move to their correct positions as dictated by Column C's sequence.

This technique effectively bypasses the need for complex scripting or manual reorganization,

offering a dynamic and reliable method for cross-column alignment. It is crucial to remember that the MATCH function is case-insensitive in Google Sheets by default, meaning it will correctly find matches even if the capitalization differs, provided the strings are otherwise identical. Furthermore, ensuring the range used for the lookup (Column C) is locked using absolute references (e.g., `C2:C11`) is essential for correct formula propagation.

Detailed Syntax and Application of the MATCH Function

To implement the indexing strategy, we must construct the MATCH function formula precisely. This formula must be applied to the first relevant cell in our temporary Column B. Since our data starts in row 2, we will input the formula into cell **B2**. The function syntax is straightforward:
`=MATCH(search_key, range, search_type)`.

In our scenario:

The **search_key** is the value we are looking for in the target list, which is the corresponding cell in Column A (e.g., `A2`).

The **range** is the entire target list (Team List 2), which is `C2:C11`. The dollar signs ensure that this range remains constant when the formula is dragged down.

The **search_type** must be 0 (zero). This critical argument specifies that we are looking for an **exact match**. Using 1 or -1 would be appropriate for sorted numerical data but is entirely unsuitable for textual matching sequences like team names.

Therefore, we type the following formula into cell **B2**:

`=MATCH(A2, C2:C11, 0)`

After entering the formula in B2, you must apply it to every remaining cell in Column B that corresponds to the rows of data. This is typically achieved by clicking and dragging the small square handle (the fill handle) at the bottom right corner of cell B2 down to cell B11. Column B will then be populated with numerical indices indicating the required sort order.

The resulting column is the numerical manifestation of the desired sequence, ready to be used as the primary sorting key. Notice how the indices in Column B directly reflect the row number of the corresponding team name within the C2:C11 range, creating a perfect numerical blueprint for alignment.

B2 ∇ | fx =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				

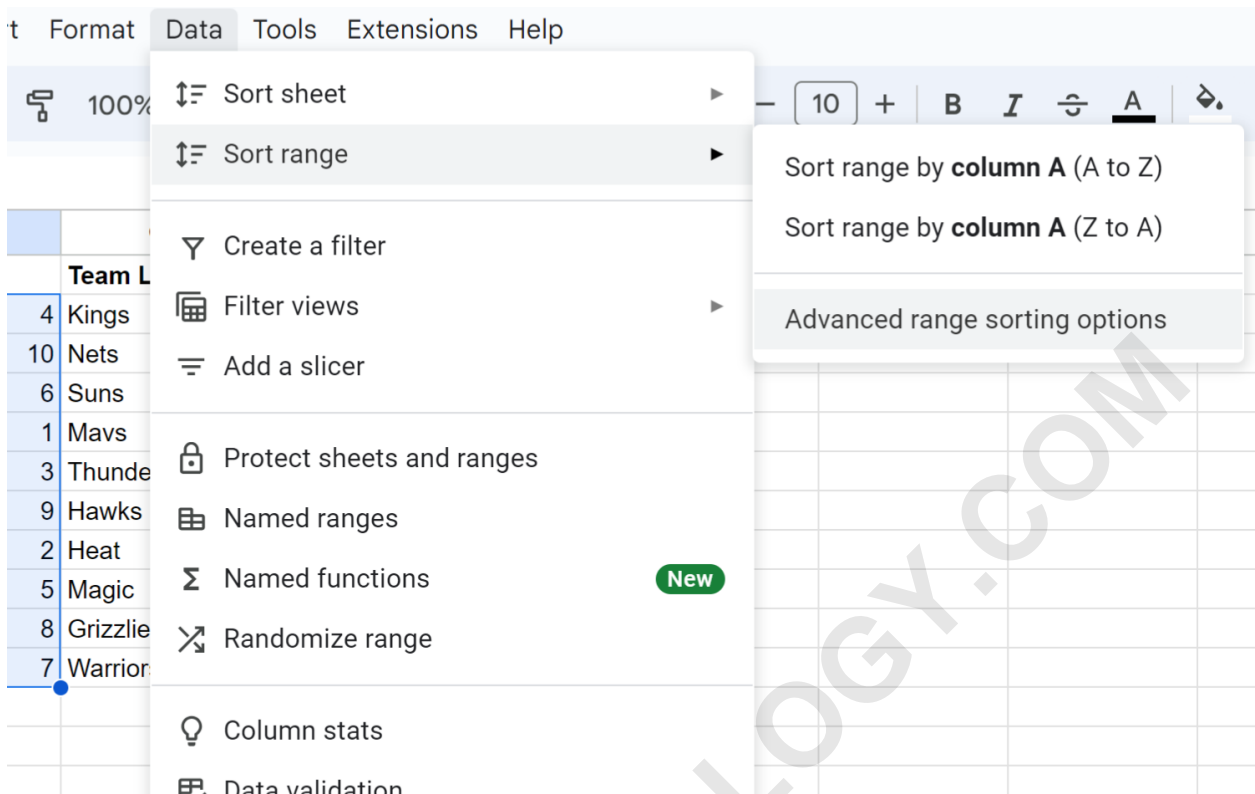
Note: Comprehensive documentation detailing the precise behavior and parameters for the [MATCH function](#) is available in the official [Google Sheets Help Center](#).

Executing the Advanced Range Sort Operation

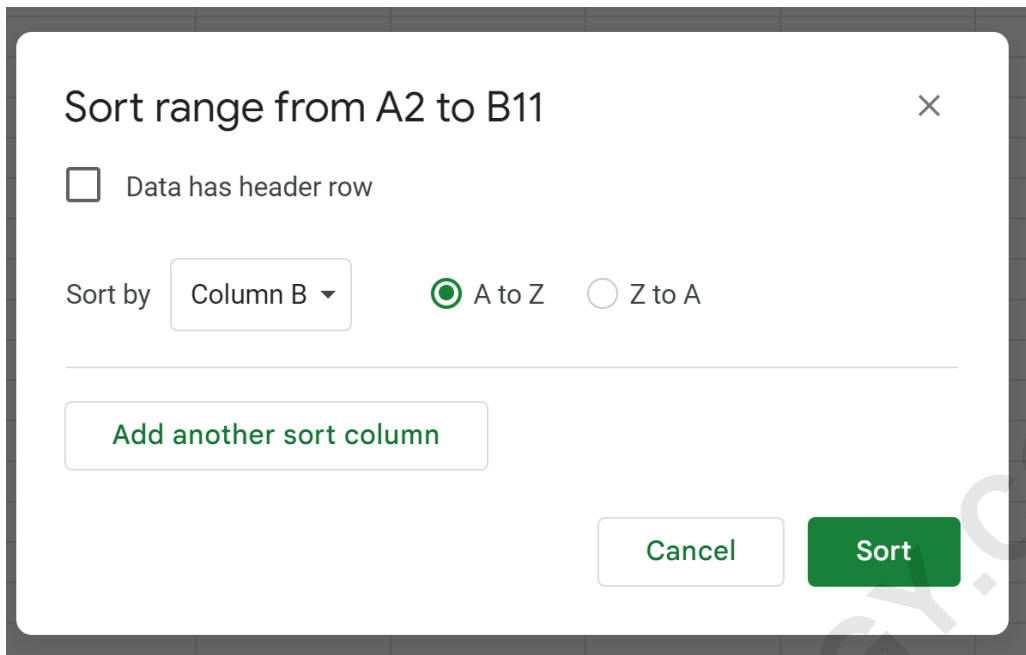
With the numerical index column now successfully populated, the final step involves using the advanced sorting features within Google Sheets to apply this index key to the combined data set. This step is where the synchronization is physically executed, moving the items in Column A (and their associated index values in Column B) into the target sequence.

To begin, accurately define the range that needs to be sorted. It is absolutely crucial to highlight both the column containing the data to be sorted (Column A) and the calculated index column (Column B). In our example, this corresponds to the cell range **A2:B11**. Failing to include Column B in the selection would mean that the sorting key is not available to the operation, and the process would fail.

Once the range is highlighted, navigate through the Google Sheets menu ribbon: click the **Data** tab, then select **Sort range**. Because we need to specify a sorting key other than the default leftmost column (A), we must choose **Advanced range sorting options** from the subsequent submenu. This action opens a dedicated dialog box designed for multi-column sorting criteria.



Inside the advanced sorting window, we must specify the exact criterion for organization. The primary instruction is to **Sort by** Column B. Since Column B contains positive integer indices (1, 2, 3, etc.), sorting this column in ascending order (A to Z, or smallest to largest) will naturally organize the rows according to the sequence defined in the target list (Column C). Select "Column B" from the drop-down menu and ensure the sorting direction is set to "A to Z" or "Ascending." Finally, click **Sort** to execute the operation.



Finalizing the Sort and Data Cleanup

Upon clicking the **Sort** button, Google Sheets instantly rearranges the rows within the selected range (A2:B11). The team names in Column A are now reordered based on the numerical sequence provided by the MATCH function results in Column B. Crucially, because Column A and Column B were sorted together as a single unit, the items in Column A have moved to exactly the position required to align them perfectly with the corresponding items in Column C.

	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				

The final confirmation of success is observing that the order of the team names in Column A is now identical to the sequence in Column C. The process of using the index column has achieved the desired matching sort. At this stage, the temporary index column (Column B) has served its purpose and is no longer needed. To maintain a clean and professional spreadsheet layout, it is highly recommended to delete the values in Column B.

Simply select the entire range of values in Column B (B2:B11) and press the Delete key. The final output provides two distinct lists, Column A and Column C, that are perfectly matched row-for-row, organized according to the order established by the target list.

	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				

Notice that the team names in each list are now in the exact same order. This method provides a robust and non-destructive way to harmonize data across multiple columns based on a predefined sequence.

Practical Applications of Matched Sorting

This advanced sorting technique has significant practical implications across various data sets and analytical tasks. One common application is reconciliation. If you receive updated pricing information (List C) that needs to align with an existing product catalog (List A) where the rows have become misaligned due to manual changes or accidental sorting, this method ensures instantaneous synchronization without requiring manual copy-pasting.

Another powerful use case involves pivot table preparation or VLOOKUP functionality. VLOOKUPs and similar functions operate most efficiently and reliably when the lookup array is consistently ordered. Using the Sort Range technique ensures that two or more source columns used in different lookups are always presented in the same sequence, reducing the risk of calculation errors and improving formula stability, particularly in complex financial models or inventory management systems.

In summary, generating a temporary sorting index using the MATCH function provides a highly efficient and reliable method for solving the problem of misaligned data columns in Google Sheets. By treating the numerical index as the primary sorting key, we leverage the power of advanced

sorting features to achieve perfect synchronization, guaranteeing the integrity of paired data relationships across your spreadsheet.

ARABPSYCHOLOGY.COM