

How do you sort numbers in Excel?

Authored by
stats writer

November 18, 2025

RECOMMENDED CITATION

stats writer (2025). *How do you sort numbers in Excel?*. PSYCHOLOGICAL SCALES.
Retrieved from <https://scales.arabpsychology.com/?p=95447>

While Excel provides straightforward built-in tools for basic data sorting, these methods require manual execution and physically rearrange the dataset. For advanced data manipulation, such as creating a secondary, dynamically sorted list without altering the original source data, or for use within more complex formulas, leveraging dedicated functions is essential. This method provides a powerful, non-destructive way to generate sorted arrays directly within your worksheet, allowing for greater flexibility and automation in your reports and analyses. We will explore how to use a combination of the SMALL function and the LARGE function, paired with the ROWS function, to achieve dynamic numerical sorting.

The core concept behind this technique is using array-ranking functions (SMALL and LARGE) to extract values sequentially. These functions require two arguments: the data array (the range of numbers you wish to sort) and a rank number (k), which specifies which smallest or largest value you want to retrieve. Crucially, the ROWS function is employed to automatically increment this rank number as the formula is dragged down the column, thereby generating the complete sorted sequence dynamically. This approach guarantees that the results instantly update if the underlying source data changes, offering a powerful alternative to static sorting methods.

Core Formulas for Dynamic Numerical Sorting

To implement this advanced numerical sorting, you will utilize slightly different formulas depending on the direction required--either smallest to largest or largest to smallest. Both formulas rely on absolute referencing for the data range to ensure it remains fixed when copying the formula, while using a carefully structured relative reference within the ROWS function to generate the necessary sequential rank (k).

The following syntax structures are universally applicable for sorting a single column of numerical data. Assume the data range is defined as $\$A\$2:\$A\13 for these examples. The key distinction lies in whether you invoke the SMALL function or the LARGE function, which dictates the resulting sort order.

Formula 1: Sort Numbers in Ascending Order (Smallest to Largest)

=SMALL(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))

Formula 2: Sort Numbers in Descending Order (Largest to Smallest)

=LARGE(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))

These formulas are designed to be entered into the first cell of your results column (e.g., cell B2) and then copied down for the entire length of the original dataset. Both structures reference the

numerical data contained within the range **A2:A13**. The power of this solution lies in the collaboration between the ranking function and the addressing mechanism supplied by the ROWS function, ensuring that each cell in the result column pulls the next sequential value.

Understanding the Mechanism: SMALL/LARGE and ROWS

To truly master this technique, one must appreciate how the internal components of the formula interact, especially the use of mixed and absolute references. The primary functions, SMALL or LARGE, are static ranking tools; they only return a value based on a static rank 'k'. For example, if k=3, they return the 3rd smallest or 3rd largest value. The challenge is generating k=1, k=2, k=3, and so on, automatically as the formula is copied down.

This is where the ROWS function comes into play. The syntax ROWS(\$A\$2:A2) is critical. The starting cell, \$A\$2, is absolutely referenced using dollar signs, meaning it will never change. The ending cell, A2, is relatively referenced (no dollar signs). When this formula is entered into cell B2, the range is \$A\$2:A2, which contains exactly one row, resulting in ROWS() returning 1. This means the formula in B2 asks for the 1st smallest (or largest) value.

When the formula is dragged down one row to cell B3, the relative reference updates: the range becomes \$A\$2:A3. This range covers two rows (A2 and A3), causing ROWS() to return 2. The formula in B3, therefore, requests the 2nd smallest (or largest) value. This mechanism continues down the column, creating a self-incrementing rank counter that drives the entire dynamic sorting process without the need for manual array entry or VBA scripting. This ingenious use of referencing is the foundation of many advanced Excel techniques.

Setting Up the Data and Visualization

Before demonstrating the specific examples, it is useful to visualize the source data we will be manipulating. The following image displays the unsorted list of numerical values contained in column A, specifically within the range **A2:A13**. Our goal is to generate the sorted results in column B, keeping column A intact.

	A	B	C	D	E	F
1	Numbers					
2	14					
3	19					
4	30					
5	18					
6	12					
7	9					
8	4					
9	45					
10	47					
11	35					
12	13					
13	16					
14						
15						
16						
17						
18						

This initial setup is crucial for understanding the outcome. Column A remains the static source, while column B will contain the dynamic output generated by the formulas. Note that the numbers are deliberately mixed to illustrate the effectiveness of the sorting functions across various magnitudes. We will now proceed to implement the SMALL function for ascending order.

Example 1: Sorting Numbers in Ascending Order

To sort the numbers from the smallest value to the largest value, we must employ the SMALL function. The logic dictates that we first retrieve the 1st smallest value, then the 2nd smallest, and so on, until the entire list is exhausted. We begin the process by typing the following formula directly into cell B2, which is the starting point for our sorted output column:

=SMALL(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))

Upon hitting Enter, cell B2 immediately returns the absolute smallest value found within the range A2:A13. This happens because, as explained previously, the ROWS function evaluates to 1 in this initial cell. The formula is then ready to be propagated down the column. We can use the fill handle (the small square at the bottom right corner of the selected cell) and drag it down to cell B13, corresponding to the total number of entries in the source data.

As the formula is dragged down, the relative component of the ROWS function increments the rank k (from 1 to 12). For instance, cell B5 will effectively execute the formula as `=SMALL(A2:A13, 4)`, returning the 4th smallest number. The resulting output in column B displays the exact same numbers as column A, but arranged in perfect ascending order (smallest to largest), demonstrating the effectiveness of this dynamic sorting method in Excel.

	A	B	C	D	E	F
1	Numbers	Ascending Order				
2	14	4				
3	19	9				
4	30	12				
5	18	13				
6	12	14				
7	9	16				
8	4	18				
9	45	19				
10	47	30				
11	35	35				
12	13	45				
13	16	47				
14						
15						
16						
17						
18						

As clearly illustrated in the resulting table above, Column B now successfully displays the numerical data from column A arranged dynamically in ascending order. This outcome confirms that the formula correctly utilizes the rank supplied by the ROWS function. The first cell in column B represents the 1st smallest value in column A, the second cell represents the 2nd smallest value in column A, and so forth, until the largest number is reached at the end of the range.

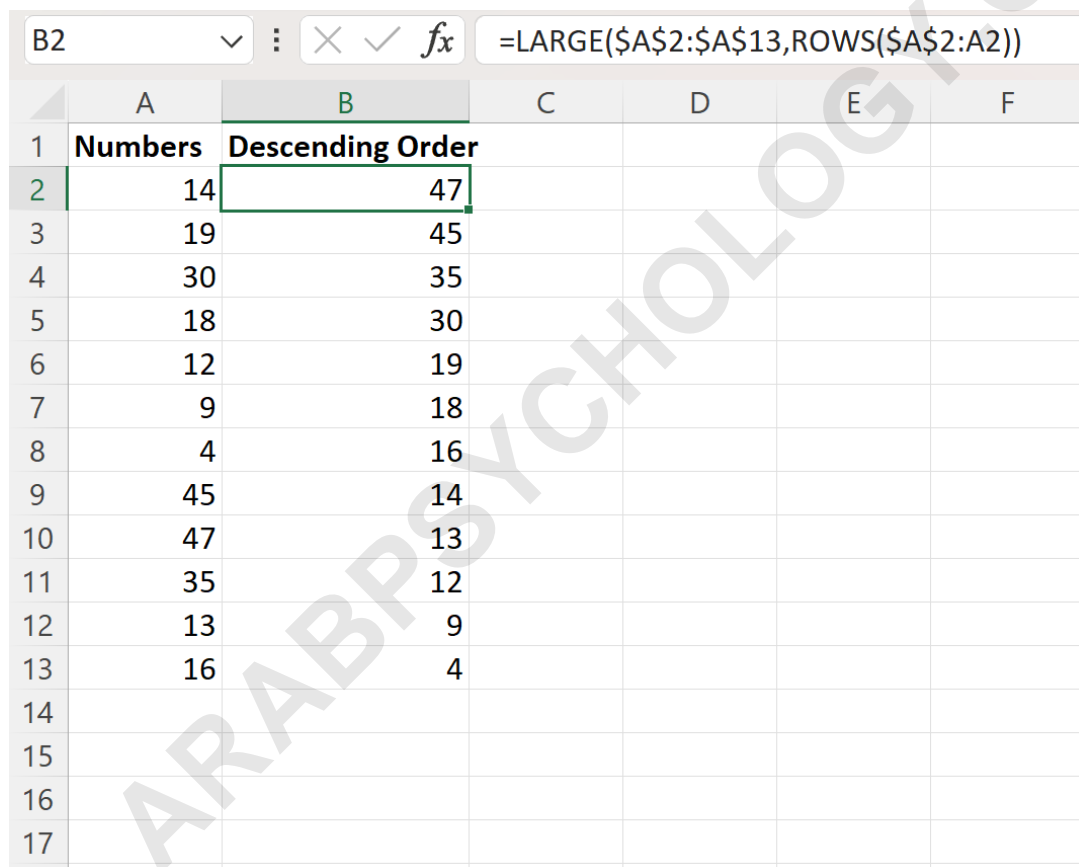
Example 2: Sorting Numbers in Descending Order

To reverse the sorting direction--from the largest value to the smallest value--we substitute the SMALL function with the LARGE function. The structure of the rest of the formula, including the critical `A2:A13` range reference and the rank generator `ROWS(A2:A2)`, remains identical. This consistency highlights the elegance of this method: the change of a single function name dictates the entire sort orientation.

We input the following formula into cell B2 to initiate the descending order sort. Since the ROWS function returns k=1 in B2, the LARGE function will first return the absolute largest value from the specified range A2:A13:

=LARGE(\$A\$2:\$A\$13,ROWS(\$A\$2:A2))

After entering the formula in B2, we click and drag the fill handle down to B13 to populate the remainder of the results column. As the formula is copied down, the incremental rank k ensures that Excel sequentially pulls the 2nd largest, 3rd largest, 4th largest, and so on, until the entire dataset is arranged in descending order. This demonstrates how easily the direction of the dynamic sorting can be controlled simply by choosing between the two ranking functions.



	A	B	C	D	E	F
1	Numbers	Descending Order				
2	14	47				
3	19	45				
4	30	35				
5	18	30				
6	12	19				
7	9	18				
8	4	16				
9	45	14				
10	47	13				
11	35	12				
12	13	9				
13	16	4				
14						
15						
16						
17						

The output above confirms the successful implementation of the LARGE function for descending order. Column B now presents the numbers from column A sorted from the highest value to the lowest value. Just as in the ascending example, the integrity of the original data in column A remains untouched. The first value displayed in column B is the 1st largest value from column A, the second value is the 2nd largest, and this pattern continues precisely down the column, fulfilling the requirement for a dynamic, formula-driven sort.

Advanced Considerations and Limitations

While the combination of SMALL/LARGE and ROWS is incredibly powerful for numerical sorting, it is important to understand its limitations and how to handle common issues, particularly duplicate values. Unlike traditional Excel sorting which maintains the original relative order of duplicate entries (stable sort), these ranking functions are not inherently stable.

If your source data contains duplicate numbers, this formula will return the duplicate values consecutively, but it cannot differentiate between them based on their original position. For instance, if the number 5 appears twice in column A, the formula will return 5 twice. If you require a stable sort (where secondary criteria, like row number, determines the order of duplicates), you would need to augment this formula using advanced techniques involving Excel's array functions (e.g., combining SMALL, ROW, and COUNTIF) to create a unique sort key for each number. However, for simple numerical sorting, the formulas demonstrated here are efficient and sufficient.

Furthermore, this methodology is specifically tailored for numerical data. Sorting text strings dynamically requires different functions, typically involving helper columns and combinations of MATCH, INDEX, and COUNTIF functions, or leveraging the advanced FILTER and SORT functions available in newer versions of Excel. Nevertheless, for ensuring dynamic numerical sorting without altering the source data, the use of SMALL or LARGE combined with the relative referencing of the ROWS function remains the definitive and cleanest formula-based approach.