

Excel: Find First Negative Value in Range

Authored by
Mohammed loot

November 10, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Excel: Find First Negative Value in Range*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15243>

In the world of **data analysis**, pinpointing the exact moment a metric transitions from positive achievement to a [negative value](#) is often a crucial requirement. For professionals utilizing [Excel](#) for financial modeling or performance tracking, modern functionality provides highly efficient, automated solutions to locate this critical threshold. This comprehensive guide will walk you through the process of combining the powerful [XLOOKUP function](#) with the mathematical [SIGN function](#) to instantly identify the very first occurrence of a negative entry within a designated data array.

The methodology presented here represents a significant upgrade over older, more cumbersome techniques. By leveraging the advanced capabilities of modern Excel, particularly its inherent ability to handle **dynamic arrays**, we bypass the need for complex, nested formulas involving combinations like `INDEX` and `MATCH`. This streamlined approach simplifies the logic: we effectively create a temporary lookup array that categorizes every number by its sign, allowing the lookup function to instantly locate the desired result with minimal syntax overhead. This technique is indispensable for analysts seeking speed and reliability.

The Optimized Formula for Finding the First Negative Value

To ensure the swift and accurate detection of the first negative number within any numerical data set, the following formula structure should be employed. This elegant solution exploits the logical transformation provided by the [SIGN function](#), which converts the range of raw data into a simplified array consisting only of 1s (positive), 0s (zero), and -1s (negative). This simplification makes the subsequent lookup process for [XLOOKUP](#) exceptionally straightforward.

=XLOOKUP(-1,SIGN(B2:B13),A2:B13)

This robust, single-cell formula is specifically structured to search for the value **-1**--which represents a [negative number](#)--within the transformed lookup array generated from the primary data [range B2:B13](#). Once the first match is successfully identified, the formula executes the retrieval of corresponding data from the **return array**, which is defined as **A2:B13**. This configuration is particularly beneficial as it allows the user to retrieve not only the negative value itself but also crucial associated contextual data, such as a date stamp, transaction ID, or, as demonstrated in our example, an employee's name from the adjacent column.

To implement this successfully, understanding the three core arguments is vital. The first argument, **-1**, is the precise **lookup_value** we instruct Excel to search for. The second argument, **SIGN(B2:B13)**, is the engine that dynamically generates the necessary **lookup_array**. Finally, **A2:B13** defines the **return_array**, clearly dictating the scope of information that must be extracted and displayed once the first match is detected. This sophisticated yet concise technique provides unparalleled efficiency when identifying critical performance thresholds across extensive datasets.

Practical Application: Case Study Setup

To fully appreciate the practical utility of this formula, let us consider a typical corporate scenario focused on tracking employee financial performance. Imagine a dataset meticulously recording the total profits generated by various team members over a fiscal reporting period. Our primary business objective is to rapidly identify the specific employee who first registers a loss--indicated by a [negative value](#)--in their recorded net profits, necessitating immediate review or intervention.

The following illustration presents this hypothetical dataset, which is clearly structured with the employee names residing in Column A and their corresponding Net Profits listed in Column B. This arrangement is common in many business reporting structures and perfectly demonstrates the formula's ability to cross-reference data:

	A	B	C	D	E	
1	Employee	Net Profits				
2	Andy	100				
3	Bob	120				
4	Chad	84				
5	Doug	13				
6	Eric	-21				
7	Frank	39				
8	Greg	-20				
9	Henry	-80				
10	Isaac	14				
11	John	19				
12	Kendall	30				
13	Luke	-12				
14						
15						
16						
17						
18						

A manual inspection of the **Net Profits** column reveals a mix of positive financial results and several instances of losses. Our goal transcends simple identification; we must locate the employee who appears earliest in the list whose profit figure is below zero. This task is essential for timely business management and performance diagnostics, powerfully demonstrating the necessity of automating this lookup process using advanced Excel functions like [XLOOKUP](#).

Relying on the traditional method of manually scanning hundreds or even thousands of data rows is highly inefficient, time-consuming, and significantly prone to human error. By leveraging this

sophisticated array formula, the process is streamlined to a single, instantaneous calculation, immediately directing the analyst's attention to the precise row of data containing the first instance of a negative profit entry.

Executing the XLOOKUP Formula to Find the First Match

To initiate the search for the first negative profit figure, we simply input the structured formula into an accessible empty cell, such as **D2**. Given that the profit figures are contained within the [range B2:B13](#) and the corresponding employee identifiers are in **A2:A13**, we must carefully define our lookup and return ranges accordingly.

We accurately enter the complete formula into cell **D2**:

=XLOOKUP(-1,SIGN(B2:B13),A2:B13)

	A	B	C	D	E	F
1	Employee	Net Profits		Employee	Net Profits	
2	Andy	100		Eric	-21	
3	Bob	120				
4	Chad	84				
5	Doug	13				
6	Eric	-21				
7	Frank	39				
8	Greg	-20				
9	Henry	-80				
10	Isaac	14				
11	John	19				
12	Kendall	30				
13	Luke	-12				
14						
15						
16						
17						

Upon execution, the formula dynamically returns two distinct values: the employee's name and their specific negative profit amount. Because we utilized **A2:B13** as the final argument (the return array), the [XLOOKUP function](#) successfully extracts and displays the corresponding data simultaneously from both the **Employee** and **Net Profits** columns for the very first row where the

profit criterion was met. This immediate, two-column output is exceptionally valuable for direct reporting, verification, and subsequent data manipulation.

We can confidently perform a manual inspection of the original data to verify the formula's accuracy. The output should definitively identify the entry for "Eric" as the first chronological instance of a recorded loss, thereby confirming the formula's effectiveness in scanning the array sequentially from the top down. This verification step underscores that the combined power of the [SIGN function](#) and XLOOKUP correctly identified the required performance threshold.

	A	B	C	D	E
1	Employee	Net Profits		Employee	Net Profits
2	Andy	100		Eric	-21
3	Bob	120			
4	Chad	84			
5	Doug	13			
6	Eric	-21			
7	Frank	39			
8	Greg	-20			
9	Henry	-80			
10	Isaac	14			
11	John	19			
12	Kendall	30			
13	Luke	-12			
14					
15					
16					

Customizing the Output Range

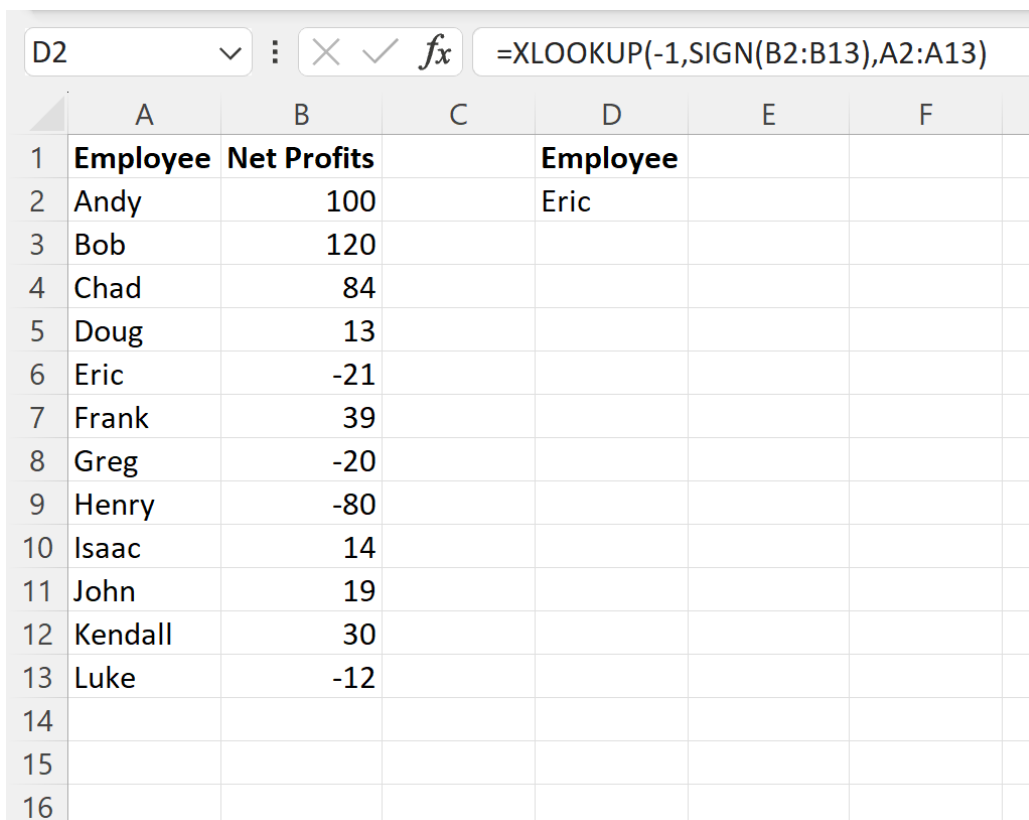
While retrieving both the primary identifier (employee name) and the financial figure is often the default requirement, certain analytical tasks may necessitate retrieving only a single, specific piece of identifying information. The inherent flexibility of the [XLOOKUP function](#) allows us to easily narrow the returned result by precisely adjusting the **return_array** argument.

In our initial formula execution, we utilized the [range A2:B13](#) as the return array to secure two columns of data. If, however, our specific objective is strictly to retrieve only the name of the employee associated with the first negative profit, we must meticulously modify the formula to limit the return array to column A exclusively.

We execute this adjustment by altering the formula in cell **D2** to the following structure:

=XLOOKUP(-1,SIGN(B2:B13),A2:A13)

By changing the final argument from a two-column [range](#) (**A2:B13**) to a single column (**A2:A13**), we explicitly instruct [Excel](#) to return only the corresponding value located within the **Employee** column. This demonstrates the critical importance of the third argument in dictating both the scope and the nature of the final output displayed to the user.



	A	B	C	D	E	F
1	Employee	Net Profits		Employee		
2	Andy	100		Eric		
3	Bob	120				
4	Chad	84				
5	Doug	13				
6	Eric	-21				
7	Frank	39				
8	Greg	-20				
9	Henry	-80				
10	Isaac	14				
11	John	19				
12	Kendall	30				
13	Luke	-12				
14						
15						
16						

The resulting output now clearly displays only the employee's name, "Eric," confirming that the lookup successfully identified the correct row based on the negative profit criteria and extracted the data precisely from the defined single-column return range. This ability to exercise granular control over the output dimension renders [XLOOKUP](#) an exceptionally versatile tool for complex and customized reporting requirements.

	A	B	C	D	E	F
1	Employee	Net Profits		Net Profits		
2	Andy	100		-21		
3	Bob	120				
4	Chad	84				
5	Doug	13				
6	Eric	-21				
7	Frank	39				
8	Greg	-20				
9	Henry	-80				
10	Isaac	14				
11	John	19				
12	Kendall	30				
13	Luke	-12				
14						
15						
16						
17						

Deconstructing the Formula: How XLOOKUP and SIGN Function Together

Achieving a deeper conceptual understanding of the internal mechanics of this formula is essential for maximizing its potential applications and effectively troubleshooting any potential errors. Let us revisit the structured formula utilized to find the first [negative value](#):

=XLOOKUP(-1,SIGN(B2:B13),A2:B13)

The ingenuity of this technique rests entirely on the nested utilization of the [SIGN function](#). The primary role of [SIGN](#) is to rigorously evaluate every numerical entry within the specified range (**B2:B13**) and return a simple, corresponding numerical indicator of its sign. This transformation is systematic and follows three distinct rules for any given value X:

If X is **positive** (greater than zero), SIGN(X) returns **1**.

If X is **zero**, SIGN(X) returns **0**.

If X is **negative** (less than zero), SIGN(X) returns **-1**.

When applied across the entire [range B2:B13](#), the [SIGN function](#) effectively generates a synthetic array composed exclusively of 1s, 0s, and -1s. This new array then functions as the necessary **lookup_array** for [XLOOKUP](#). Since we have fixed the **lookup_value** to **-1**, XLOOKUP is

instructed to locate the first occurrence of -1 in this generated array. Because XLOOKUP processes arrays sequentially (top-to-bottom), it inherently guarantees that the very first match it finds corresponds precisely to the initial negative value encountered in the original profit column. This sophisticated yet elegant combination eliminates the need for complex Boolean logic or cumbersome array formulas that required CSE entry in older [Excel](#) versions, making data analysis faster, cleaner, and far more accessible.

Summary and Best Practices

The strategic combination of the [XLOOKUP function](#) and the [SIGN function](#) provides the most efficient and modern solution available for rapidly pinpointing the initial deviation into negative territory within any structured dataset. This method is markedly superior to legacy lookup techniques because it is natively designed to handle dynamic array operations, dramatically simplifying the required formula syntax and improving computational speed.

When implementing this powerful solution, it is imperative to ensure strict alignment between your arrays. Specifically, your **lookup_array** (the numerical range being tested, e.g., B2:B13) and your **return_array** (the range containing the data you intend to retrieve, e.g., A2:B13) must be vertically aligned and contain an identical number of rows. Any misalignment between these two ranges will invariably lead to critical calculation errors, typically manifesting as a **#VALUE!** or **#REF!** error. Additionally, remember that this formula is specifically tailored to detect the first instance of a value less than zero; if your goal shifts to finding the first positive value, you must simply change the **lookup_value** from **-1** to **1**.

This technique demonstrates remarkable flexibility and can be seamlessly adapted to a diverse range of analytical scenarios beyond traditional financial data. Whether you are tracking fluctuations in temperature, monitoring inventory depletion levels, or evaluating any performance metric where identifying the first occurrence of a specific sign change is crucial, this straightforward and powerful Excel formula offers a reliable and instantaneous solution.

Further Excel Resources

The following resources offer additional tutorials explaining how to perform other common and advanced operations in [Excel](#), further complementing your skills in advanced data lookup and manipulation: