

# How to Use an IF Function with Range of Values in Google Sheets

Authored by  
**stats writer**

November 21, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Use an IF Function with Range of Values in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99286>

The IF function in Google Sheets is a fundamental tool for conditional analysis, allowing users to execute one action if a specified condition evaluates to TRUE and a different action if it evaluates to FALSE. Its power is significantly amplified when dealing with a range of values, such as determining if a specific numerical entry falls within predefined upper and lower boundaries or if a text string exists within a defined cell selection. Mastering this application enables complex data categorization and automated decision-making processes within your spreadsheets.

When working with ranges, the standard IF function often needs to be nested with other logical functions, most commonly the AND function, to enforce multiple criteria simultaneously. For instance, to verify if a number is truly within a range, we must check two conditions: is the value greater than or equal to the lower limit, AND is the value less than or equal to the upper limit? This combination ensures that the conditional logic is precise and handles boundary conditions correctly, leading to powerful and versatile calculations across large datasets.

## Understanding Conditional Logic for Data Ranges

Working with ranges requires specific formulas tailored to the type of data being evaluated--whether you are searching for a specific text string across a group of cells or verifying that a numerical entry satisfies multiple limits. Google Sheets provides optimized approaches for both scenarios, ensuring efficient processing without requiring complex array formulas in many cases. These methods rely on combining the core logic of the IF statement with functions designed to handle aggregation or multiple criteria checks.

The structure of the conditional test is paramount. If we are checking for the presence of a value within a range of cells, we utilize a counting mechanism that tallies the occurrences of the search term. If the count exceeds zero, the condition is met. Conversely, when checking if a single numerical value falls within a prescribed numeric interval, we often employ a more direct method using Boolean logic, which converts multiple conditional tests into numerical values that can be easily evaluated by the IF function.

## Methods for Implementing IF Functions with Ranges

To effectively manage conditional logic across a range of values in Google Sheets, two primary methodological approaches are utilized based on whether you are analyzing textual presence across multiple cells or evaluating a single numeric value against boundaries. The following formulas summarize these two powerful techniques:

You can use the following formulas to create an IF function with a range of values in Google Sheets:

### Method 1: Creating an IF Function with a Range of Cells (Text Presence)

**=IF(COUNTIF(A2:A11,"Pacers")>0, "Exists", "Does Not Exist")**

## Method 1: Checking for Text Presence using COUNTIF

This first method is designed to determine if a specific text string or value is present anywhere within a designated range of cells. This is achieved by nesting the COUNTIF function inside the logical test of the IF statement. The COUNTIF function counts the number of cells within the specified range, **A2:A11**, that match the provided criteria, which in this example is the string "Pacers."

The conditional logic then checks whether the count returned by COUNTIF is greater than zero (>0). If the count is greater than zero, it confirms that the search term "Pacers" exists at least once in the range, and the IF function returns the value specified for the TRUE outcome ("Exists"). Conversely, if the count is exactly zero, the search term is absent, and the function returns the FALSE outcome ("Does Not Exist"). This method is highly effective for auditing data presence across large columns or rows.

For this formula, if the text string "Pacers" exists anywhere within the cell range **A2:A11**, then the function successfully returns "Exists." If the string is not found in that range, the function returns "Does Not Exist."

## Method 2: Evaluating Numeric Ranges Using Boolean Logic

The second method addresses the requirement of testing a single cell value against two separate numeric boundaries--a lower limit and an upper limit. Instead of using the dedicated AND function, this technique leverages implicit Boolean logic by multiplying the results of two separate comparisons. In spreadsheet environments, a TRUE condition is often interpreted as the numeric value 1, and a FALSE condition as 0.

The formula checks two conditions: first, whether the value in cell **B2** is greater than or equal to 95 ( $B2 \geq 95$ ), and second, whether the value in **B2** is less than or equal to 105 ( $B2 \leq 105$ ). When these two comparisons are multiplied together, the resulting product will only equal 1 (TRUE) if and only if BOTH conditions are TRUE ( $1 * 1 = 1$ ). If either condition is FALSE (0), the product will be 0, signaling that the number is outside the desired range. This elegant use of arithmetic simplifies the conditional test.

## Method 2: Creating an IF Function with a Range of Numeric Values

**=IF(((B2>=95)\*(B2<=105))=1, "Yes", "No")**

For this formula, if the numeric value in cell **B2** is inclusively between 95 and 105 (i.e.,  $95 \leq B2 \leq 105$ ), the resulting output is "Yes." If the value falls outside of this specified interval, the function returns "No."

## Setting up the Demonstration Dataset

To illustrate the practical application of both methods, we will use a sample dataset containing information about fictional sports teams. This dataset includes the Team Name and the Points scored by that team. The goal of the following examples is to apply the conditional formulas to this data to automatically evaluate team presence and score performance against predetermined criteria. This provides a clear, visual context for how the formulas interact with real-world spreadsheet data.

The following examples show how to use each formula in practice with the following dataset in Google Sheets:

	A	B	C	D	E
1	<b>Team</b>	<b>Points</b>			
2	Mavericks	99			
3	Heat	94			
4	Nets	104			
5	Hornets	108			
6	Kings	114			
7	Warriors	98			
8	Pelicans	90			
9	Knicks	93			
10	Pacers	95			
11	Cavs	100			
12					
13					
14					
15					
16					
17					
18					
19					
20					

## Practical Application: Checking for Cell Existence (Example 1)

In this first example, we utilize Method 1 to check for the presence of a specific team name within

the list spanning the range A2 through A11. This scenario is common when performing data validation or conducting quick checks to confirm if an expected entry is included in a master list. Our objective is to search for the team "Pacers" and return a definitive status based on whether it appears in the list.

We can type the following formula into cell **D2** to return "Exists" if the team name "Pacers" exists in the range **A2:A11** or to return "Does Not Exist" if the team name does not exist in the range:

**=IF(COUNTIF(A2:A11,"Pacers")>0, "Exists", "Does Not Exist")**

Since the team "Pacers" is indeed listed in row A6 of our dataset, the COUNTIF function will return a value of 1. As 1 is greater than 0, the IF condition is satisfied (TRUE), and the function delivers the string "Exists." If we were to change the search criterion to a team name not present in the list, such as "Lakers," the COUNTIF function would return 0, resulting in the FALSE output, "Does Not Exist."

The following screenshot shows how to use this formula in practice, confirming the presence of the specified team:

	A	B	C	D	E
D2				=IF(COUNTIF(A2:A11,"Pacers")>0, "Exists", "Does Not Exist")	
1	<b>Team</b>	<b>Points</b>		<b>Pacers Exists?</b>	
2	Mavericks	99		Exists	
3	Heat	94			
4	Nets	104			
5	Hornets	108			
6	Kings	114			
7	Warriors	98			
8	Pelicans	90			
9	Knicks	93			
10	Pacers	95			
11	Cavs	100			
12					
13					
14					
15					
16					
17					
18					
19					
20					

The formula successfully returns "Exists" because the string "Pacers" occurs at least once within the specified data range **A2:A11**.

### Practical Application: Testing Against Numeric Boundaries (Example 2)

Example 2 focuses on evaluating the numeric scores in Column B against a predefined target range of values (95 to 105). This application is crucial for tasks like grading, performance classification, or flagging data points that fall outside acceptable thresholds. Unlike Example 1, which checked a static range for presence, this example applies the logic dynamically to each row to assess its individual score.

We apply Method 2's formula, which uses the multiplicative technique for compound conditional checks, starting in cell D2:

**=IF(((B2>=95)\*(B2<=105))=1, "Yes", "No")**

After entering the formula in cell D2, we can then utilize the fill handle feature to drag this formula down and apply it to each corresponding cell in column D, automatically adjusting the row references (B2, B3, B4, etc.) as it propagates:

	A	B	C	D
D2				$=IF(((B2 \geq 95) * (B2 \leq 105)) = 1, "Yes", "No")$
1	<b>Team</b>	<b>Points</b>		<b>Points Between 95 and 105?</b>
2	Mavericks	99		Yes
3	Heat	94		No
4	Nets	104		Yes
5	Hornets	108		No
6	Kings	114		No
7	Warriors	98		Yes
8	Pelicans	90		No
9	Knicks	93		No
10	Pacers	95		Yes
11	Cavs	100		Yes
12				
13				
14				
15				
16				
17				
18				
19				

## Decoding the Numeric Range Formula

The resulting output in column D clearly categorizes each team's points relative to the specified range. The logic applied in the formula can be summarized as follows for every row evaluated:

If the value in the Points column (Column B) is inclusively between 95 and 105, the formula returns "Yes," indicating the score is within the target range.

If the value in the Points column is *not* between 95 and 105 (i.e., less than 95 or greater than 105), the formula returns "No," indicating the score falls outside the target range.

It is important to understand the role of the multiplication symbol (\*) within this boolean structure. When comparing two conditions using multiplication, Google Sheets treats the operation like an AND function. For the entire conditional test to return TRUE (1), both logical statements must be

individually TRUE. If we had used addition (+) instead of multiplication (\*), the formula would behave like an OR statement, returning TRUE if at least one condition were met (since  $1 + 0 = 1$ ). The use of the multiplication operator is therefore critical in this context, signaling to Google Sheets that **both** the lower limit and the upper limit must be satisfied simultaneously.

Finally, note that the resulting output values ("Yes" and "No") are entirely customizable. You can modify the last two arguments in the IF function to return any text string, numerical value, or even another formula, depending on the required action when the condition is met or unmet. For example, you could return "In Threshold" or "Needs Review" instead of simple Boolean responses.

**Note:** The multiplication symbol (\*) in the IF function effectively tells Google Sheets that both conditional statements must be evaluated as TRUE for the resulting logical test to be TRUE and thus return the "Yes" argument.