

# How to Return a Value Based on a Number Range in Excel

Authored by  
**stats writer**

February 24, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Return a Value Based on a Number Range in Excel*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=132490>

**Microsoft Excel** is an exceptionally versatile and powerful application that serves as a cornerstone for modern **data analysis**, financial modeling, and administrative organization. One of the most common yet impactful tasks a user can perform is the conditional categorization of numerical data based on specific boundaries. By leveraging logical functions, users can instruct the software to evaluate whether a numeric value falls within a designated range and subsequently return a customized result. This capability is essential for managing large datasets where manual inspection is impractical, providing a mechanism to automate decision-making processes and ensure high levels of accuracy in reporting. Whether you are tracking academic grades, managing inventory levels, or analyzing athletic performance metrics, mastering range-based logic is a fundamental skill for any proficient user of a **spreadsheet** environment.

## Excel: If Number is Between Range Then Return Value

### The Fundamentals of Conditional Logic in Microsoft Excel

To effectively determine if a number resides within a specific numerical span, one must understand the interplay between the IF function and the AND function. The **IF function** acts as a gatekeeper, evaluating a **Boolean expression** to decide which of two possible outcomes to display. However, a standard **IF function** can only evaluate a single condition at a time. To check if a number is between two different points, we must simultaneously verify that it is greater than the lower bound and less than the upper bound. This is where the **AND function** becomes indispensable, as it allows for the nesting of multiple criteria within a single logical test, returning a value of "TRUE" only if every individual condition within the parentheses is met.

When constructing these formulas, it is vital to distinguish between exclusive and inclusive ranges. An exclusive range omits the boundary numbers themselves, whereas an inclusive range includes them. This distinction is controlled by the **relational operators** used within the formula. For instance, using the "greater than" (>) and "less than" (<) and "less than or equal to" (<=) ensure that the boundary values are part of the successful criteria. Choosing the correct operator is a critical step in maintaining **data integrity** and ensuring that your **data analysis** yields the expected results without edge-case errors.

The following formulas provide a standardized approach to checking ranges within **Microsoft Excel**. By following these established methods, you can return specific text strings, such as "Yes" or "No," or even trigger secondary calculations if the condition is satisfied. This logical framework is the building block for more complex operations, such as nested **IF statements** or integration with the **VLOOKUP** and **XLOOKUP** functions for more dynamic data retrieval strategies.

## Method 1: Evaluating Ranges Using Exclusive Boundaries

The exclusive method is utilized when a value must strictly fall inside a window without touching the limits of that window. This is common in scenarios where the boundary numbers represent a shift into a different category. To implement this, we nest an AND function inside the logical test argument of the IF function. This structure ensures that both logical constraints are satisfied before the formula returns the "True" result. If either condition fails, the formula defaults to the "False" result, providing a clear binary output for the user.

**=IF(AND(B2>20, B2<30), "Yes", "No")**

In this specific example, the formula examines the data stored in a particular cell reference, which in this case is **B2**. The **AND function** checks two specific conditions: first, whether the value is strictly greater than 20, and second, whether the value is strictly less than 30. If a number such as 25 is present in **B2**, the formula returns "Yes." However, if the cell contains the number 20 or the number 30, the formula will return "No," because these values do not satisfy the "greater than" or "less than" requirements. This precision is essential in **statistical analysis** where specific intervals must be strictly observed.

The exclusive range formula is often the preferred choice when dealing with continuous data or when the boundaries themselves represent distinct "out of bounds" markers. By clearly defining these limits, the user can avoid the common mistake of double-counting values that might otherwise fall into two overlapping categories. Utilizing this method enhances the clarity of the **spreadsheet** and ensures that the logic is transparent to other users who may need to audit or update the workbook in the future.

## Method 2: Evaluating Ranges Using Inclusive Boundaries

In many practical business and scientific applications, the boundaries of a range are considered part of the valid dataset. This is known as an inclusive range. To capture these boundary values, we modify our **relational operators** to include the equals sign. This ensures that if the value in the cell reference matches the limit exactly, it is still categorized within the desired range. This method is frequently used in grading scales, tax brackets, and performance tiering.

**=IF(AND(B2>=20, B2<=30), "Yes", "No")**

When applying this inclusive formula to **B2**, **Microsoft Excel** evaluates if the number is greater than or equal to 20 AND less than or equal to 30. Unlike the previous method, the values 20 and 30 will now trigger a "Yes" response. This is a common requirement in **data analysis** when the target range represents a comprehensive set of acceptable values. The inclusion of the equals

sign provides a broader net, ensuring that no data points are inadvertently excluded due to a strict boundary definition.

Using inclusive boundaries is particularly helpful when working with integers or discrete data points where the limit itself is a significant milestone. For example, if a "Passing" grade starts exactly at 60, an inclusive formula ensures that a student who earns exactly 60 is marked as "Passing." Understanding when to toggle between exclusive and inclusive logic is a hallmark of sophisticated **spreadsheet** design and allows for highly nuanced data processing.

## Practical Implementation with Dataset Examples

To better visualize how these logic strings function in a real-world environment, consider a dataset containing the performance metrics of professional basketball players. In this scenario, we are tracking the "Points" scored by various individuals in a series of games. Our objective is to identify which players scored within a specific target range to assist in performance evaluation. This type of **data analysis** is standard in sports analytics and helps coaches and managers make informed decisions based on empirical evidence.

	A	B	C	D	E	F
1	<b>Team</b>	<b>Points</b>				
2	Mavs	22				
3	Spurs	19				
4	Rockets	20				
5	Kings	15				
6	Warriors	30				
7	Nets	34				
8	Lakers	28				
9	Thunder	24				
10	Blazers	17				
11	Jazz	19				
12						
13						
14						
15						
16						

The image above illustrates our initial dataset, with Column A containing player names and Column B containing their respective points. By applying our formulas to Column C, we can create a dynamic status report that updates automatically if the point values are changed. This

automation is one of the primary benefits of using **Microsoft Excel**, as it reduces the need for manual data entry and minimizes the risk of human error during the calculation process.

By structuring the data in this manner, we can quickly apply filters or **conditional formatting** to the results in Column C, allowing us to highlight high-performers or identify players who may need additional training. The ability to transform raw numbers into actionable "Yes/No" or "Pass/Fail" indicators is a vital component of efficient information management.

### Example 1: Executing the Exclusive Range Test

In our first practical example, we will apply the exclusive range formula to our basketball dataset. We want to determine if a player's points are strictly between 20 and 30. We start by entering the formula into the first data row of Column C, specifically in cell **C2**, which corresponds to the first player's score in **B2**. This allows the **IF function** to evaluate the points and return the appropriate text string based on our criteria.

**=IF(AND(B2>20, B2<30), "Yes", "No")**

After inputting the formula, we utilize the **fill handle**--a small square at the bottom-right corner of the active cell--to drag the logic down through the rest of the column. This action uses **relative cell references** to automatically adjust the formula for each row (changing **B2** to **B3**, **B4**, and so on). This is a core feature of **Microsoft Excel** that enables the rapid processing of hundreds or thousands of rows in a matter of seconds.

	A	B	C	D	E
1	<b>Team</b>	<b>Points</b>	<b>Between 20 and 30 (Exclusive)</b>		
2	Mavs	22	Yes		
3	Spurs	19	No		
4	Rockets	20	No		
5	Kings	15	No		
6	Warriors	30	No		
7	Nets	34	No		
8	Lakers	28	Yes		
9	Thunder	24	Yes		
10	Blazers	17	No		
11	Jazz	19	No		
12					
13					
14					
15					
16					

As observed in the updated image, the results in Column C reflect the exclusive nature of our test. You will notice that any player who scored exactly 20 or 30 points receives a "No" result. This confirms that our formula is functioning correctly, as it is only looking for values that are *greater than* 20 and *less than* 30. This level of specificity is crucial for data sets where "almost" hitting a target does not count as meeting the criteria.

## Example 2: Executing the Inclusive Range Test

In the second example, we shift our focus to an inclusive range. This is often more appropriate for performance tiers where hitting the exact boundary is considered a success. By adjusting our logical operators, we tell **Microsoft Excel** to include the numbers 20 and 30 in the "Yes" category. This subtle change in the Boolean expression can significantly alter the outcome of our **data analysis**.

**=IF(AND(B2>=20, B2<=30), "Yes", "No")**

Once again, we apply the formula to the top of our column and propagate it downward. The **Microsoft Excel** engine re-evaluates each score against the updated inclusive parameters. This flexibility allows users to experiment with different boundary conditions to see how they impact the overall results of the dataset, a process often referred to as sensitivity analysis.

	A	B	C	D	E
1	<b>Team</b>	<b>Points</b>	<b>Between 20 and 30 (Inclusive)</b>		
2	Mavs	22	Yes		
3	Spurs	19	No		
4	Rockets	20	Yes		
5	Kings	15	No		
6	Warriors	30	Yes		
7	Nets	34	No		
8	Lakers	28	Yes		
9	Thunder	24	Yes		
10	Blazers	17	No		
11	Jazz	19	No		
12					
13					
14					
15					
16					

The resulting Column C now shows a "Yes" for players who scored exactly 20 or 30 points. This demonstrates the practical difference between the two methods. In a business context, this could be the difference between a salesperson receiving a bonus or missing out. By mastering these formulas, you gain granular control over how your data is interpreted and presented to stakeholders.

## Optimizing Your Excel Workflow with Logical Functions

Beyond simple range checking, these logical functions can be expanded and combined with other tools to create robust automated systems. For instance, you could replace the "Yes" and "No" text in the formula with further calculations, such as multiplying the points by a bonus multiplier if they fall within the range. This allows for dynamic financial modeling where outputs change instantly as input data is updated.

Furthermore, understanding how to use AND functions within an **IF statement** is a gateway to learning other logical operators like **OR** and **NOT**. These functions allow you to handle even more complex scenarios, such as checking if a number is within Range A OR Range B, or checking if a number is NOT within a specific exclusion zone. Combining these logic gates enables the creation of highly sophisticated **data analysis** templates that can handle a wide variety of business rules.

To maintain a clean and professional **spreadsheet**, it is also recommended to use named ranges for your boundaries. Instead of typing "20" and "30" directly into your formula, you could reference cells that contain these values. This makes your formulas easier to read and allows you to update the boundaries for the entire sheet by changing just two cells, rather than editing every formula individually. This practice is a hallmark of professional **Microsoft Excel** development and significantly improves the maintainability of your work.

## Expanding Your Knowledge of Excel Operations

The ability to return specific values based on numerical ranges is just one of many essential skills for navigating **Microsoft Excel** effectively. As you become more comfortable with conditional logic, you may find it useful to explore related topics that can further enhance your productivity and analytical capabilities. Professional proficiency in **spreadsheet** software involves a continuous process of learning and applying new techniques to solve complex problems.

**Conditional Formatting:** Use the same logical principles to automatically change cell colors or fonts based on their values.

**Nested IF Statements:** Learn how to check for multiple ranges (e.g., Low, Medium, High) within a single formula.

**Data Validation:** Prevent users from entering numbers that fall outside of your required ranges in the first place.

**Array Formulas:** Perform range checks across entire arrays of data simultaneously for more advanced **data analysis**.

By integrating these methods into your daily routine, you will transform **Microsoft Excel** from a simple table-making tool into a powerful engine for insight and automation. The following tutorials provide in-depth explanations on how to perform other common and advanced tasks within the software, ensuring you have the resources necessary to tackle any data challenge that comes your way.