

# Learn How to Use IF and AND Functions to Check Number Ranges in Excel

Authored by  
**Mohammed looti**

November 9, 2025

## RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Use IF and AND Functions to Check Number Ranges in Excel*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=14955>

## Mastering Conditional Range Checks in Excel

In advanced data analysis, a fundamental task is determining whether a specific data point, such as a [number](#), successfully falls within a predefined numerical [range](#) or interval. In [Excel](#), this complex logical evaluation is accomplished efficiently by strategically combining the [IF function](#) with the [AND function](#). This powerful nested structure allows users to execute sophisticated [Boolean logic](#) checks that require two or more conditions to be simultaneously true. Specifically, for a number to be "in range," it must satisfy two criteria: it must be greater than the lower boundary AND less than the upper boundary. If this dual condition is met, the formula returns a specified success value (e.g., "Yes"); otherwise, it returns the alternative result (e.g., "No"). Mastering this technique is essential for automated data filtering, conditional reporting, and building robust spreadsheet models.

When defining these parameters, it is critical to distinguish between two primary methodologies: the **exclusive range check** and the **inclusive range check**. The difference hinges entirely on how the formula treats the boundary numbers themselves. An exclusive check strictly excludes the boundary values, whereas an inclusive check incorporates them as valid members of the set. Understanding this subtle but profound difference is vital for ensuring the precision of your data analysis and compliance with required mathematical or business constraints. Each method relies on a specific set of [logical operators](#) (such as  $>$ ,  $<$ ,  $>=$ , or  $<=$ ) to meticulously define the edges of the interval being evaluated. This precision in scope definition is the cornerstone of accurate data classification when working with extensive numerical datasets.

### Implementing an Exclusive Range Check (Open Interval)

The **exclusive range check** is designed for scenarios where the boundary values must be strictly excluded from the valid outcome. Consider a requirement to identify values strictly between 20 and 30; in this case, scoring exactly 20 or exactly 30 must result in a failure or "No" outcome. To enforce this strict criterion, we must use the strict inequality operators: greater than ( $>$ ) and less than ( $<$ ). The formula structure leverages the [AND function](#) to mandate that both criteria--being larger than the lower bound AND smaller than the upper bound--are satisfied simultaneously. If even one of these conditions is not met, the **AND function** yields FALSE, causing the outer [IF function](#) to return the alternative result.

This approach forms the foundation of exclusive range validation in [Excel](#), ensuring that the defined limits are not included in the successful range determination. Mathematically, this corresponds to checking membership within an open interval, often denoted as  $(20, 30)$ . The following standard formula illustrates how to check if a value residing in cell **B2** is strictly between 20 and 30, returning "Yes" only if the constraints are entirely satisfied.

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

The formula meticulously evaluates two independent conditions: first, whether the value in cell **B2** exceeds 20, and second, whether that same value is less than 30. Only when both comparisons evaluate to TRUE does the **IF function** execute the "Yes" outcome. This methodology provides a precise and unambiguous mechanism for exclusion checks, which is indispensable in regulatory or grading scenarios where specific cutoffs must be strictly observed, such as minimum age requirements or performance tiers that do not include the precise boundary scores.

## Implementing an Inclusive Range Check (Closed Interval)

In direct contrast to the exclusive approach, the **inclusive range check** is employed whenever the boundary values themselves must be treated as valid members of the numerical set. For example, if the required range is defined as 20 to 30, a score of exactly 20 and a score of exactly 30 must both be considered successful matches. To achieve this necessary inclusivity, we utilize the corresponding **logical operators**: greater than or equal to ( $\geq$ ) and less than or equal to ( $\leq$ ). This variation is ubiquitous in applications like defining inventory thresholds, tiered payment structures, and academic grading systems where the endpoints of the interval carry equal weight to the internal values.

As with the exclusive method, the inclusive check relies on the coordinating power of the **AND function** to simultaneously confirm two conditions: that the target **number** meets or exceeds the lower limit AND that it is less than or equal to the upper limit. Failure to meet either inclusive condition results in a FALSE output from the **AND** statement. This methodology mathematically defines a closed interval, symbolized as  $[20, 30]$ . The corresponding formula for an inclusive check is structured as follows:

**=IF(AND(B2 $\geq$ 20, B2 $\leq$ 30), "Yes", "No")**

This specific formula determines if the value present in cell **B2** falls within the defined **range** of 20 through 30, granting membership to both endpoints. The inclusion of the equality component in the operators radically shifts the analysis compared to the exclusive check, providing essential flexibility for many data classification tasks. When developing these inclusive logical statements, meticulous attention to the precise configuration of the operators is paramount, as a minor error in placement or selection can invalidate calculations across extensive datasets, leading to misclassification of important boundary data points.

## Practical Demonstration: Preparing the Dataset

To clearly demonstrate the operational distinctions and utility of both inclusive and exclusive range

checks, we will apply both formulas to a representative sample dataset. This dataset tracks performance metrics, specifically the points scored, for a group of basketball players. Our objective is to rapidly identify which players meet a defined scoring criteria--falling within the 20 to 30 point [range](#)--by deploying these conditional formulas within [Excel](#).

The structure of our sample data includes two columns: the Player Name and the Points Scored (Column B). We will introduce a third column, labeled "In Range?", which will host the results generated by our logical conditional checks. This setup provides an immediate, visual assessment of which numerical entries satisfy the defined logical parameters. Reviewing and understanding the initial dataset structure is always the crucial first step in any analytical process, as it dictates the specific structure of the required [cell referencing](#) within our formulas. We must ensure that our calculations accurately target the points data in Column B while populating the results in Column C.

	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						

For the subsequent examples, we will exclusively focus our formulas on evaluating the points values listed in Column B. The following steps will detail the precise entry of the formulas into Column C, starting at cell **C2**. We will then leverage Excel's efficient drag-and-fill functionality to automatically apply the complex logical evaluation across all player entries. This practical workflow underscores how the combination of **IF** and **AND** functions automates complex decision-making across potentially massive lists of data, significantly reducing the time required compared to manual verification. A clear grasp of relative [cell referencing](#) is fundamental to successfully

replicating the formula's logic down multiple rows.

## Example 1: Analyzing Exclusive Range Results

In this detailed first scenario, we deploy the **exclusive range check** formula to identify players who scored strictly more than 20 points but strictly less than 30 points. We are essentially establishing a specific middle-tier performance bracket where the endpoints (20 and 30) are intentionally excluded. This is highly useful for isolating intermediate performers who surpass a minimum threshold but do not reach the highest defined level. We input the formula into cell **C2**, ensuring it references the points value in **B2**, structured precisely as follows:

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

Once the formula is accurately entered into cell **C2**, the next step is to efficiently apply this logic throughout the entire dataset column. This is achieved by utilizing the fill handle--the small square at the corner of the cell--and dragging it down to cover the corresponding rows in Column C. Excel's built-in relative referencing mechanism automatically adjusts the [number](#) of the row reference (B2 automatically becomes B3, B4, and so forth). This streamlined process demonstrates a core advantage of using spreadsheet software for data manipulation, enabling instantaneous conditional evaluation across potentially thousands of records.

	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					

A careful examination of the resulting output clearly illustrates the consequence of using exclusive operators. The formula successfully returns "Yes" only for scores like 22, 25, and 28, confirming that they fall strictly within the 20 and 30 boundaries. Crucially, note the outcomes for entries that scored exactly 20 or 30. Because the formula demanded scores be *greater than 20* and *less than 30*, these boundary values fail the test and are consequently assigned a result of "No." This outcome rigorously emphasizes why selecting the correct [logical operators](#) (>, <) is paramount when defining criteria that must entirely exclude boundary conditions. The underlying principle of [Boolean logic](#) requires that both constituent conditions linked by the **AND** operator must be satisfied simultaneously for the overall outcome to be TRUE.

## Example 2: Analyzing Inclusive Range Results

Our second comprehensive example shifts the focus to the **inclusive range check**. In this scenario, the objective is to correctly identify all players whose scores reside within the 20 to 30 point [range](#), explicitly ensuring that those scoring exactly 20 or exactly 30 are also successfully included in the "Yes" category. This methodology defines a complete performance tier from the absolute minimum acceptable score to the maximum. We modify the structure by incorporating the greater than or equal to ( $\geq$ ) and less than or equal to ( $\leq$ ) operators within the [IF function](#) and **AND function** combination. The revised formula, which is entered into cell **C2**, is structured as follows:

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

After inputting this formula, we repeat the efficient step of using the fill handle to copy the inclusive logic down the column, applying the check across all remaining player scores. The fundamental difference in the operation here stems from the logical flexibility provided by the equality component embedded in the operators. This adjustment means the formula is now testing for membership within a mathematically closed interval, guaranteeing that marginal success cases--those scores landing precisely on the boundary--are not mistakenly excluded from the desired classification.

	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					

Reviewing the final output highlights the crucial deviation from Example 1. While all values between 20 and 30 correctly yield "Yes," the scores that are exactly 20 or exactly 30 now also return "Yes." This successful inclusion occurs because the conditional statement checks if the points value is *greater than or equal to 20 AND less than or equal to 30*. This precise inclusion of the boundary values delivers a complete and accurate portrayal of performance within the specified criteria, demonstrating the high level of precision available when correctly applying inclusive [logical operators](#) in [Excel](#) for critical data classification tasks.

## Expanding Beyond Binary Range Checks

While the foundational [IF\(AND\(\)\)](#) structure is the most direct and clearest method for returning a binary result (such as "Yes" or "No"), these core logical checks serve as a springboard for significantly more complex data manipulation. For instance, instead of merely returning a success or failure text value, the final argument of the [IF function](#) can be replaced by another sophisticated calculation, a separate formula, or even a series of nested logical tests. This flexibility facilitates multi-tiered evaluation criteria, enabling users to classify numerical scores into detailed buckets like "Low," "Medium," or "High" based on successive, tiered range checks.

For scenarios requiring the classification of scores across many distinct, contiguous ranges--such

as complex grading scales with five or more tiers (A, B, C, D, F), where each tier represents a specific score [range](#)--relying on extensive nested **IF** statements becomes cumbersome and error-prone. In such cases, utilizing powerful lookup functions like **VLOOKUP** or **XLOOKUP**, specifically by setting the range lookup argument to TRUE (to enable approximate matching), proves far more efficient and manageable. However, the streamlined **IF(AND())** combination remains the ideal choice for quick, unambiguous pass/fail assessments or two-category range checks. Developing a robust understanding of how [Boolean logic](#) translates into these spreadsheet functions is absolutely crucial for creating resilient and highly scalable data models.

To further enhance your data analysis capabilities beyond simple range checking, consider exploring these related [Excel](#) tutorials and functions:

Using the **OR function** for conditional checks where only one of several conditions must evaluate to true.

Implementing **conditional formatting** to visually highlight cells based on whether their values fall within specified numerical ranges.

Creating dynamic reports or charts that automatically filter or summarize data based on the logical outcomes of range tests.

Applying the aggregate functions like **SUMIFS** or **COUNTIFS** to efficiently total or count data entries that satisfy multiple criteria, including specified numerical ranges.

By fully mastering the versatility of the **IF** and **AND** functions, you equip yourself with indispensable tools for precise data segmentation, automated decision-making, and sophisticated reporting within virtually any spreadsheet environment.