

Excel Formula: If Greater Than But Less Than

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Excel Formula: If Greater Than But Less Than*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94695>

The Power of Nested Conditional Logic in Excel

The ability to perform complex logical tests is fundamental to effective Excel formula usage and advanced data analysis. One of the most common requirements in quantitative analysis is determining if a specific data point falls within a predefined range--that is, if it is simultaneously greater than a lower bound and less than an upper bound. This task is accomplished in Excel by combining the powerful **IF function** with the versatile **AND function**, resulting in a robust structure often referred to as the 'If Greater Than But Less Than' formula. Mastering this combination significantly enhances your capability to automate decision-making processes within your spreadsheets.

This specific type of nested formula structure is critical when generating conditional statements where multiple criteria must be satisfied for a TRUE result. Unlike a simple comparison (e.g., $A > 10$), checking for a range requires two separate logical evaluations to occur simultaneously. For instance, if you are analyzing inventory levels, you might need to identify products whose stock count is above the reorder threshold (greater than X) AND below the maximum safety stock level (less than Y). Without the **AND function**, achieving this dual-condition test would necessitate excessively long and complicated nested IF statements, compromising efficiency and clarity.

Understanding how to properly structure this comparison is vital for anyone who regularly works with numerical values, dates, or structured data sets in Excel. Whether you are filtering sales data to find transactions between \$1,000 and \$5,000, classifying student grades based on numerical scores, or identifying time periods within a specific calendar range, this composite formula provides an elegant and straightforward solution. It transforms manual data auditing into an automated, scalable process, solidifying its place as an indispensable tool in the advanced Excel user's toolkit.

Deconstructing the IF and AND Functions

To master the combined formula, one must first thoroughly understand the roles of its two primary components: the IF function and the AND function. The IF function is the decision-maker; it checks a condition and returns one value if that condition is TRUE, and another value if it is FALSE. Its basic syntax is `=IF(logical_test, value_if_true, value_if_false)`. The `logical_test` is the core argument that must resolve to either TRUE or FALSE.

The AND function, conversely, is the condition aggregator. It takes multiple logical tests as arguments and only returns TRUE if **all** of those tests evaluate to TRUE. If even one test fails (returns FALSE), the entire AND function returns FALSE. Its syntax is `=AND(logical1, , ...)`. This function is perfectly suited to handle range checking, as it allows us to simultaneously verify two distinct conditions (greater than X, and less than Y) on a single cell reference.

When we nest the AND function inside the logical test argument of the IF function, we create a

powerful mechanism. The AND function performs the dual criteria check, and its resulting TRUE or FALSE output feeds directly into the IF function. If the AND function returns TRUE (meaning both criteria were met), the IF function executes the `value_if_true` argument. If the AND function returns FALSE (meaning at least one criterion failed), the IF function executes the `value_if_false` argument. This nesting structure is the foundation for creating precise conditional statements in Excel that manage complex boundary requirements.

The Core Syntax: If Greater Than But Less Than

The standard structure used to implement the 'If Greater Than But Less Than' logic is remarkably clean and efficient. It relies on clearly defining the cell to be tested, the minimum boundary, and the maximum boundary. The formula is constructed to ensure that the cell value meets both inequality requirements simultaneously. Below is the archetypal structure used to evaluate if a value is within a non-inclusive range:

The following Excel formula demonstrates how to check if the value contained in a particular cell is greater than a specified lower limit but strictly less than a defined upper limit:

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

In this specific example, the formula initiates a check to see if the content of cell **B2** is greater than the number 20 AND simultaneously less than the number 30. If both conditions are TRUE, the cell returns "Yes"; otherwise, it returns "No". The logical tests within the **AND function** are separated by commas, ensuring a clear delineation of the criteria being evaluated.

Crucially, notice the use of the strict inequality operators: greater than (>) and less than (<). If the requirement were to include the boundary values (e.g., greater than or equal to 20 AND less than or equal to 30), the operators would need to be changed to `>=` and `<=`, respectively. Understanding the impact of these operators is essential for accurate range filtering when dealing with numerical values, as including or excluding a boundary point can drastically alter the results for edge cases.

Setting Up the Practical Example Data

To illustrate the practical application of this powerful nested formula, let us consider a common scenario involving sports statistics. Suppose we are tasked with identifying players whose recent performance metrics fall within a specified desirable scoring range--a range that is high enough to show contribution but not so high as to skew the team's average significantly. We require a method to quickly flag players whose point totals are greater than 20 but less than 30.

We begin with a structured dataset within Microsoft Excel, as shown below. This table contains the names of various basketball players, their respective teams, and the total number of points they

scored during a recent period. The goal is to create a new column that automatically generates a binary response (Yes/No) based on the range check condition.

The input data is structured as follows, where Column B contains the key metrics we intend to evaluate:

	A	B	C	D	E
1	Player	Points			
2	Andy	22			
3	Bob	14			
4	Chad	17			
5	Doug	30			
6	Eric	35			
7	Frank	18			
8	Greg	11			
9	Henry	19			
10	Isaac	24			
11	John	16			
12	Kendall	40			
13	Luke	33			
14					
15					
16					

Our specific objective is to determine whether each individual value listed in the **Points** column (Column B) satisfies the criteria: **Points > 20 AND Points < 30**. This task requires the precise deployment of the combined Excel formula, starting in the adjacent cell, C2.

Applying the Formula Step-by-Step

The implementation process is straightforward, beginning with the construction of the formula in the first target cell (C2). Since we are evaluating the value in cell B2, we replace the general reference in our syntax breakdown with this specific cell address. The structure ensures that the two criteria are logically linked via the AND function.

To achieve the desired filtering--checking if the point total is greater than 20 but less than 30--we type the following exact formula into cell **C2** to do so:

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

Once entered, Excel evaluates cell B2 (which contains the value 22). The first logical test, `B2>20`, is TRUE (22 is greater than 20). The second logical test, `B2<30`, is also TRUE (22 is less than 30). Since both conditions within the AND function are met, the entire logical test resolves to TRUE, and the IF function returns the value specified in the `value_if_true` argument, which is "Yes".

The final crucial step involves applying this formula dynamically to the rest of the dataset. Instead of manually retyping the formula for every row, we utilize Excel's fill handle (the small square at the bottom-right corner of cell C2). By clicking and dragging this formula down to each row in column C, the relative cell reference B2 automatically adjusts to B3, B4, B5, and so on, applying the precise range check to every player's point total in the column. This efficient process ensures consistency and accuracy across the entire table, allowing for rapid generation of complex conditional statements.

Analyzing and Interpreting the Results

After successfully dragging the formula down through column C, the spreadsheet is populated with the binary results ("Yes" or "No"), providing immediate insight into which data points satisfy the defined range criteria (greater than 20 but less than 30). This resultant column serves as a powerful filter, instantly segmenting the data based on performance thresholds.

The output clearly shows the immediate utility of this formula combination:

	A	B	C	D
1	Player	Points	Points greater than 20 but less than 30	
2	Andy	22	Yes	
3	Bob	14	No	
4	Chad	17	No	
5	Doug	28	Yes	
6	Eric	35	No	
7	Frank	18	No	
8	Greg	11	No	
9	Henry	19	No	
10	Isaac	24	Yes	
11	John	26	Yes	
12	Kendall	40	No	
13	Luke	33	No	
14				
15				
16				
17				

Column C now returns either "Yes" or "No" to indicate if each corresponding value in column B is greater than 20 but less than 30. We can examine specific rows to confirm the accuracy of the logic:

The value **22** is greater than 20 but less than 30 so the formula returns **Yes**.

The value **14** is not greater than 20 but is less than 30, so the formula fails the AND test and returns **No**.

The value **17** is not greater than 20 but less than 30, so the formula returns **No**.

A high score, such as **35** (if present), would also result in a **No**, as it fails the "less than 30" criteria.

This systematic comparison, driven by the nested IF and AND functions, provides definitive classification for every data entry, streamlining advanced data analysis and reporting processes significantly.

Alternative Applications and Noteworthy Variations

While the example above utilized the simple text outputs "Yes" and "No," the flexibility of the IF function allows for far more dynamic return values. Analysts are not restricted to binary text results; the `value_if_true` and `value_if_false` arguments can contain virtually anything, including other formulas, cell references, or mathematical calculations. For instance, instead of returning "Yes,"

you could return the value of the points themselves (B2), or perhaps apply a percentage discount calculation only if the sales figure falls within the target range.

Note: In this example, we chose to return the values "Yes" or "No". However, you can easily replace these values with alternative outputs, such as "Target Range Met" and "Out of Range," or even numerical outcomes like '1' (for true) and '0' (for false) for easy summation and aggregation later in your spreadsheet modeling.

Furthermore, careful consideration must be given to boundary conditions. As previously noted, the operators used (>, =, <=) define whether the range is inclusive or exclusive. If the requirement had been to include the boundaries (20 and 30), the formula would be written as: `=IF(AND(B2>=20, B2<=30), "Yes", "No")`. This subtle variation is critical for ensuring that edge cases, such as a player scoring exactly 20 or exactly 30 points, are categorized correctly according to the project specifications. Always confirm whether the thresholds themselves should be included in the filtering criteria when working with numerical values and range checks.

Ultimately, the nested IF-AND structure is one of the most powerful and scalable tools in the Excel formula repertoire for generating complex conditional statements that involve multiple dependencies, paving the way for sophisticated spreadsheet management and automated decision-making processes.

And so on.