

# How to Check if a Time is Between Two Times in Excel

Authored by  
**stats writer**

February 19, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Check if a Time is Between Two Times in Excel*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=131528>

The ability to accurately determine if a specific timestamp occurs within a designated range is a fundamental requirement for professional **data analysis** within Microsoft Excel. Whether you are managing complex **project management** timelines, auditing employee shift records, or organizing a high-volume **logistics** schedule, mastering temporal comparisons allows for automated and error-free reporting. In Excel, time is not merely a string of text; it is a sophisticated **data type** represented as a fractional serial number. This underlying structure enables users to apply mathematical and **logical operators** to time values just as they would with standard integers or decimals. By leveraging the power of Boolean logic, users can transform raw data into actionable insights with minimal manual intervention.

To effectively check if a given time falls between two specific boundaries, one must understand the interaction between various Excel functions. The most common approach involves the **IF function** combined with the **AND function**. For scenarios where time values are entered as text, the **TIMEVALUE function** is essential for converting those strings into the appropriate numeric format that Excel recognizes for calculation. By comparing a target time against a minimum and maximum threshold, the software can return a **TRUE** or **FALSE** result, or any custom label such as "Within Range" or "Out of Range." This methodology ensures that your **spreadsheets** remain dynamic and responsive to changes in the underlying data, providing a robust framework for any time-sensitive task.

## Excel: Methodology for Verifying Time Within a Range

To perform a high-precision check on whether a time value in a specific cell falls between two other time constraints, you can implement the following **logical formula**. This syntax is designed to be versatile, accommodating various data layouts while maintaining strict accuracy:

```
=IF(AND(C2>=MIN(A2:B2),C2<=MAX(A2:B2)),"Yes","No")
```

The technical construction of this formula serves a specific purpose: it evaluates the **timestamp** located in cell **C2** against the temporal boundaries defined in cells **A2** and **B2**. By utilizing the **AND function**, the formula ensures that both conditions--the time being greater than or equal to the start and less than or equal to the end--are met simultaneously. The **MIN** and **MAX** functions are incorporated as a safety measure to identify the lower and upper bounds of the range automatically, regardless of the order in which the user entered the start and end times. This prevents errors that might arise from human data entry inconsistencies.

The practical application of this logic is best demonstrated through a structured dataset. In professional environments, such **automation** reduces the risk of oversight and significantly speeds up the processing of thousands of rows of data. The following example provides a clear illustration of how this formula functions within a standard spreadsheet environment.

## The Logic of the IF and AND Functions

The **IF function** is the cornerstone of decision-making in conditional programming within Excel. It operates by evaluating a specific logical test and returning one value if the test is true and another if it is false. When dealing with time ranges, a single test is usually insufficient because we must verify two distinct conditions: that the event occurred after the start time and before the end time. This is where the **AND function** becomes vital. The **AND function** acts as a gatekeeper, requiring every logical statement nested within it to be true before it allows the **IF function** to return the positive result.

Furthermore, the integration of **comparison operators** such as **>=** (greater than or equal to) and **<=** (less than or equal to) allows for "inclusive" ranges. This means that if an event happens exactly at the start or end time, it is still counted as being within the range. In **data management**, defining these boundaries clearly is essential for maintaining the integrity of your results. If your specific workflow requires "exclusive" ranges, you can simply remove the equals signs to use **>** and **<** respectively, which would exclude the exact boundary times from the "Yes" result.

Advanced users often prefer this specific formula structure because of its resilience. By wrapping the range **A2:B2** inside **MIN** and **MAX**, you create a formula that is "direction-agnostic." In many real-world datasets, users might accidentally swap the "Start Time" and "End Time" columns. Without the **MIN** and **MAX** functions, a standard comparison formula would fail and return a "No" result incorrectly. This added layer of logic ensures that your **data validation** remains accurate even when the input data is slightly disorganized.

## Practical Example: Check if Time is Between Two Times in Excel

Consider a scenario where an organization needs to monitor the punctuality of specific events. Suppose we have a dataset that records a required "Start Time," a required "End Time," and the actual "Event Time" when the task was executed. The goal is to verify if the execution occurred within the allowed window.

	A	B	C	D	E
1	<b>Start Time</b>	<b>End Time</b>	<b>Event Time</b>		
2	1:15 AM	8:12 AM	7:54 AM		
3	2:12 AM	10:40 AM	10:40 AM		
4	3:15 AM	5:15 AM	5:18 AM		
5	5:19 AM	3:44 PM	5:12 AM		
6	6:55 AM	8:50 PM	8:30 PM		
7	9:30 AM	9:45 AM	10:13 AM		
8	12:15 PM	12:24 PM	12:50 PM		
9	1:45 PM	4:59 PM	5:00 PM		
10	4:50 PM	11:30 PM	11:12 PM		
11					
12					
13					
14					
15					
16					

In the image above, the data is organized into three primary columns. To efficiently process this information across multiple rows, we can apply our **logical formula** to the first data row and then propagate it throughout the rest of the column. This approach is a standard best practice in **business intelligence** for creating scalable reports.

To begin the calculation, navigate to cell **D2** and input the following formula:

```
=IF(AND(C2>=MIN(A2:B2),C2<=MAX(A2:B2)),"Yes","No")
```

Once the formula is entered, you can use the **fill handle**--the small square at the bottom-right corner of the cell--to drag the formula down to the remaining cells in column D. Excel will automatically adjust the **cell references** for each row, ensuring that the event time in row 3 is compared to the start and end times in row 3, and so on.

D2						
=IF(AND(C2>=MIN(A2:B2),C2<=MAX(A2:B2)),"Yes","No")						
	A	B	C	D	E	F
1	<b>Start Time</b>	<b>End Time</b>	<b>Event Time</b>	<b>Event Within Start and End Time?</b>		
2	1:15 AM	8:12 AM	7:54 AM	Yes		
3	2:12 AM	10:40 AM	10:40 AM	Yes		
4	3:15 AM	5:15 AM	5:18 AM	No		
5	5:19 AM	3:44 PM	5:12 AM	No		
6	6:55 AM	8:50 PM	8:30 PM	Yes		
7	9:30 AM	9:45 AM	10:13 AM	No		
8	12:15 PM	12:24 PM	12:50 PM	No		
9	1:45 PM	4:59 PM	5:00 PM	No		
10	4:50 PM	11:30 PM	11:12 PM	Yes		
11						
12						
13						
14						
15						

Upon completion, column D will provide an immediate visual confirmation of the status of each event. This **data visualization** technique allows managers to quickly identify outliers or non-compliant entries without having to manually inspect every timestamp.

## Analysis of Formula Results and Case Studies

Examining the results of the formula provides insight into how Excel handles different temporal scenarios. Let us look at specific rows from our example to understand the **logic** in action:

**Case 1:** The first event was recorded at 7:54 AM. The allowed window was between 1:15 AM and 8:12 AM. Since 7:54 AM is chronologically after the start and before the end, the **IF function** correctly returns **Yes**.

**Case 2:** The second event occurred at exactly 10:40 AM, which is the same as the end time. Because we used the **<=** operator, the formula includes the boundary and returns **Yes**.

**Case 3:** The third event time of 5:18 AM falls outside the 3:15 AM to 5:15 AM window. Since the time is later than the maximum allowed value, the **AND function** fails, and the formula returns **No**.

These results demonstrate the reliability of the **formulaic approach**. By using **Yes** and **No** as the return values, the data becomes easy to filter or count using other tools like **PivotTables** or the **COUNTIF function**. This allows for high-level summaries, such as calculating the percentage of events that occurred within their designated timeframes.

## Understanding Time as a Serial Number in Excel

To truly master time calculations, one must understand how number formatting works in the Excel environment. Excel does not see "12:00 PM" as text; it sees it as 0.5, because 12 hours is exactly half of a 24-hour day. All times are stored as decimals between 0 and 1. For instance, 6:00 AM is stored as 0.25, and 6:00 PM is 0.75. This system is what allows the **MIN function** and **MAX function** to compare times so effectively--they are simply comparing decimal values.

If your data is not behaving as expected, it is often due to the **data type**. If a time is entered as a text string, Excel may not be able to perform logical comparisons. In these instances, the **TIMEVALUE function** can be used to wrap the cell reference, converting the text back into a **serial number**. Ensuring your cells are formatted as "Time" or "General" is a vital step in the **data cleaning** process.

Furthermore, when working with dates and times together, the integer portion of the number represents the date, while the fractional part represents the time. This is why a formula designed for simple time checks might require adjustments if the timestamps include specific calendar dates. However, for most intra-day scheduling tasks, the fractional comparison provided by the **AND function** is the most efficient solution.

## Customizing the Formula for Enhanced Utility

The flexibility of the **IF function** allows you to tailor the output to meet the specific needs of your **workflow**. While "Yes" and "No" are excellent for general use, you might require more descriptive labels for corporate reporting. For example, you could modify the formula to return "On Time" or "Late," or even leave the cell blank if the condition is not met by using an empty string ("").

Consider the following customization options:

**Numerical Flags:** Use 1 and 0 instead of "Yes" and "No" if you plan to perform mathematical calculations (like **SUM**) on the results column.

**Nested Logic:** You can nest additional **IF functions** to create multiple categories, such as "Early," "On Time," and "Late."

**Conditional Formatting:** Combine the formula logic with Excel's **conditional formatting** feature to automatically highlight rows in red if they fall outside the allowed time range.

By customizing the return values, you enhance the **user experience** for anyone else viewing the spreadsheet. Clear, descriptive labels prevent confusion and make the **data analysis** more accessible to stakeholders who may not be familiar with the underlying formulas.

## Troubleshooting and Best Practices

When working with time in Excel, several common pitfalls can affect the accuracy of your **logical tests**. One frequent issue is the "Midnight Problem." If a shift starts at 10:00 PM and ends at 6:00 AM, the end time is numerically smaller than the start time because it belongs to the next day. In such cases, a simple **MIN/MAX** check might fail. To resolve this, you may need to add a 1 to the end time to signify a new **24-hour cycle** or include dates in your timestamps.

Another best practice is to ensure **data consistency**. If some times are entered with seconds (e.g., 8:12:05 AM) and others are not, the comparison might yield unexpected results. It is advisable to use the **ROUND** function if millisecond precision is causing discrepancies in your **Boolean logic** comparisons. Consistency in **formatting** ensures that your formulas produce reliable, repeatable results across different versions of Microsoft Office.

Finally, always verify that your **logical operators** match your business rules. Use **>=** and **<=** for inclusive ranges where the boundary times are acceptable. Use **>** and **<** for exclusive ranges where the event must occur strictly between the two points. Documenting these logic choices in a **data dictionary** or a hidden "Notes" sheet can be incredibly helpful for long-term project maintenance.

The following tutorials explain how to perform other common operations in Excel: