Learn to Check if a Time Falls Between Two Times in Excel

Authored by

Mohammed looti

November 10, 2025

One of the most frequent analytical challenges faced by users of Excel involves determining whether a specific event occurred within a predefined time window. This is a critical task in scheduling, log analysis, and data validation where precision regarding chronological boundaries is paramount. Fortunately, Excel provides a powerful combination of logical functions that allow us to resolve this check efficiently and reliably.

To evaluate if the time recorded in a designated cell falls within the range defined by a start time and an end time, we employ a robust nested formula structure. This structure leverages **Boolean logic** to test two conditions simultaneously, ensuring the target time is neither before the start boundary nor after the end boundary. The primary formula utilized for this task is as follows:

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

This sophisticated formula is designed to check if the time stored in cell **C2** is chronologically between the start time found in cell **A2** and the end time located in cell **B2**. Upon evaluation, the formula returns a clear textual result: either "Yes" if the condition is met, or "No" if the time falls outside the specified interval. The strategic use of the MIN function and MAX function ensures the formula works correctly regardless of whether the start time (A2) is numerically smaller than the end time (B2) or vice versa, providing flexibility that simpler comparison methods lack.

Before diving into a practical implementation, it is essential to establish a foundational understanding of how Excel internally manages time data. This context explains why the comparison operators function so effectively in this scenario.

Understanding Time Serialization in Excel

In Excel, all date and time entries are stored as serial numbers. Dates are represented by integers (the number of days elapsed since January 1, 1900), while Time values are represented as fractional portions of a 24-hour day. This serialization is crucial because it converts what we perceive as temporal data into raw numerical data that can be easily compared and manipulated using standard mathematical operators.

Specifically, time ranges from 0 (representing 12:00 AM, or 00:00) up to 0.99999... (just before the next midnight). For example, 12:00 PM (noon) is stored internally as 0.5, because it represents exactly half of a day. Similarly, 6:00 AM is stored as 0.25, representing one quarter of a day. When we compare two times, such as comparing 0.4 (9:36 AM) to 0.3 (7:12 AM), we are simply comparing two floating-point numbers. This numerical nature is what allows the greater than or equal to (`>=`) and less than or equal to (`<=`) operators to evaluate chronological order accurately.

The reliability of the time-check formula depends entirely on this serialization process. If the cell

containing the event time (C2) and the cells containing the range times (A2 and B2) are formatted correctly as time or custom date/time formats, Excel ensures that the underlying values are simple fractions. This is a powerful feature that simplifies complex time-based logic, turning a temporal problem into a straightforward numerical inequality check.

It is important to ensure consistency in data entry. If the time entries include a date component (e.g., 8/15/2023 10:00 AM), the entire serial number will be used in the comparison, which might lead to unexpected results if the goal is only to compare the time of day. For a pure time-of-day check, the date component of the serial number should typically be ignored, or the cells should contain only the time fraction (a serial number between 0 and 1).

Deconstructing the Core Time-Check Formula

The formula

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

is a testament to the nested power of Excel functions. Understanding each component is vital for troubleshooting and adaptation.

The innermost components are the MIN function and the MAX function. These functions operate on the range **A2:B2**, which contains the start time and the end time, respectively. The primary purpose of using MIN and MAX here is to dynamically define the true lower and upper bounds of the time window. Since Excel time is numerical, MIN(A2:B2) identifies the numerically smaller time (the true start), and MAX(A2:B2) identifies the numerically larger time (the true end). This approach elegantly bypasses the need for the user to guarantee that A2 is always chronologically earlier than B2, making the formula universally applicable for non-overnight ranges.

The results of the MIN and MAX calculations are then fed into the AND function. The AND function is a logical gate that requires all its arguments to be TRUE for it to return TRUE. In this case, it tests two inequalities: first, that **C2** is greater than or equal to the minimum time (`C2>=MIN(A2:B2)`); and second, that **C2** is less than or equal to the maximum time (`C2<=MAX(A2:B2)`). Only when the event time (C2) satisfies both conditions--meaning it falls exactly on or between the boundaries--does the AND function return TRUE.

Finally, the entire logical test is contained within the IF function, which serves as the output mechanism. The IF function takes the Boolean result (TRUE or FALSE) from the AND function and translates it into user-friendly text. If the logical test is TRUE, it returns "Yes." If the logical test is FALSE, it returns "No." This structure provides immediate, clear feedback on the status of each event time relative to its corresponding range.

Practical Example: Implementing the Time Range Check

To illustrate the utility of this formula, let us consider a scenario common in business operations: tracking whether specific recorded events or incidents occur during predefined operating hours or shift times. Suppose we have the following dataset in Excel that includes the start time of a shift, the end time of that shift, and the exact time an event was logged:

	A	B	C	D	E
1	**Start Time**	**End Time**	**Event Time**		
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					

Our objective is to create a fourth column that dynamically determines if the event time listed in Column C is contained within the Start Time (Column A) and End Time (Column B) for that specific row. This validation process is essential for calculating response times, verifying shift adherence, or filtering data based on operational windows. The formula must be robust enough to handle row-by-row variations in the time range.

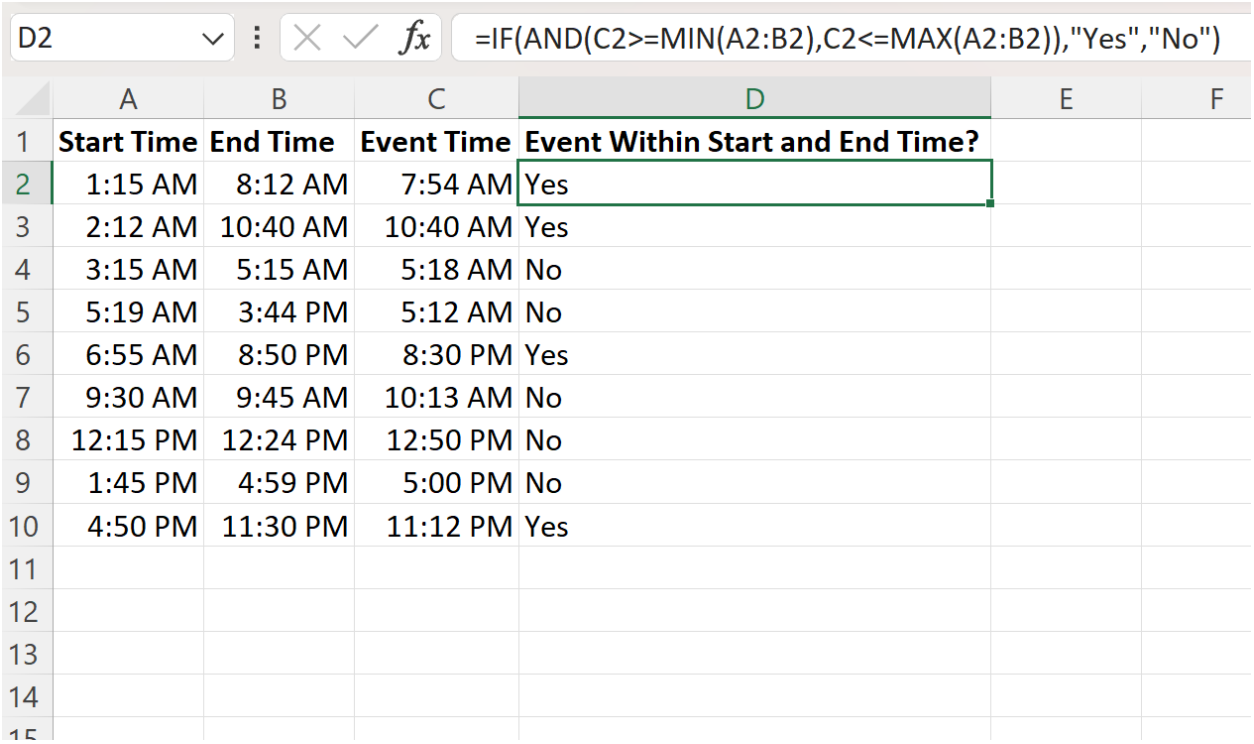
To accomplish this, we initiate the process by typing the following formula into cell **D2**, which is the first cell in our results column:

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

Once the formula is entered into D2, we can utilize Excel's powerful auto-fill feature. By clicking and dragging the fill handle (the small square at the bottom right corner of cell D2) down to the remaining cells in column D, we apply the relative reference logic of the formula across the entire dataset. This automatically adjusts the row numbers (A2, B2, C2 become A3, B3, C3, and so on),

ensuring that each row's event time is compared only against its corresponding start and end times.

The resulting table, populated with the calculated values, clearly illustrates which events occurred within the defined boundaries:



	A	B	C	D	E	F
1	**Start Time**	**End Time**	**Event Time**	**Event Within Start and End Time?**		
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						

Column D now provides a clear, decisive indicator ("Yes" or "No") of whether the recorded Event Time falls within the designated Start Time and End Time for that specific entry. This method allows for rapid assessment of large datasets without manual inspection.

Analyzing Results and Edge Cases

The results derived from the formula reveal how the logical checks handle various scenarios, including times that are exactly on the boundary and times that fall clearly outside the range. Let's examine a few key examples from the dataset to solidify our understanding:

The first event time of **7:54 AM** is numerically situated between the start time (1:15 AM) and the end time (8:12 AM). Since both inequalities within the AND function are TRUE, column D correctly returns **Yes**.

The second event time of **10:40 AM** coincides exactly with the end time of the range (2:12 AM to 10:40 AM). Because we used the greater than or equal to (`>=`) and less than or equal to (`<=`) operators, the check includes the boundaries. Consequently, the formula returns **Yes**, affirming its

inclusive nature.
The third event time of **5:18 AM** is chronologically outside the specified range of 3:15 AM to 5:15 AM. Specifically, 5:18 AM is numerically greater than the maximum time (5:15 AM). As the second condition fails, the AND function returns FALSE, and column D returns **No**.

It is paramount to recognize the significance of the inclusion criteria used in the formula. The use of **<=** and **>=** symbols ensures that if the event time is precisely equal to either the start time or the end time, the IF function will still return **Yes**. If the business requirement demands strictly exclusive boundaries (meaning the event must occur *between* the times, but not *on* them), the formula must be modified to use the strict inequality operators: `>` (greater than) and `<` (less than). This seemingly minor change dictates whether events happening at the boundary moments are included or excluded from the result set.

A crucial technical limitation to acknowledge is the handling of time ranges that span across midnight (e.g., a shift starting at 10:00 PM and ending at 6:00 AM). Since Excel treats time as a fraction between 0 and 1, a 10:00 PM time (a high fraction, close to 1) is numerically greater than a 6:00 AM time (a low fraction, close to 0). In such overnight scenarios, the simple MIN/MAX approach will fail because the maximum time (10:00 PM) is numerically greater than the minimum time (6:00 AM), but the range is defined backward. Handling overnight periods requires a more complex logical structure utilizing the OR function to check two separate conditions: either the time is greater than the start time OR the time is less than the end time.

Alternative Methods for Time Range Validation

While the MIN/MAX approach offers maximum flexibility when the order of the start and end times is uncertain, simpler methods can be employed when the dataset is highly structured and the start time is guaranteed to precede the end time.

In cases where we are absolutely certain that A2 represents the chronologically earlier time and B2 represents the later time, the formula can be simplified by removing the MIN and MAX functions:

=IF(AND(C2>=A2, C2<=B2),"Yes","No")

This streamlined formula is faster to execute and easier to read, as it involves fewer nested calculations. However, if even a single pair of start/end times is entered incorrectly (e.g., the end time is mistakenly entered before the start time), this formula will return FALSE for all event times, as no time can simultaneously be greater than a larger number and less than a smaller number. The original MIN/MAX structure acts as a safeguard against such data entry errors, making it the preferred method for general data validation.

Furthermore, instead of returning "Yes" or "No," users often require different outputs based on the

logical outcome. For instance, in a payroll context, one might want to return the text "In Shift" or "Off Shift," or even perform a calculation. The final arguments of the IF function are placeholders for any value, calculation, or text string, offering extensive customization options:

```
=IF(logical_test, value_if_true, value_if_false)
```

By replacing "Yes" and "No" with other values, the formula adapts seamlessly to various analytical requirements. For example, to return the difference in time (End Time - Start Time) if the event is within the range, or a blank cell if it is outside, the formula could be adjusted accordingly.

Summary and Best Practices for Data Validation

Checking if a time falls between two boundaries is a cornerstone of temporal data analysis in Excel. The formula utilizing the IF function, combined with the AND function and the dynamic range definition provided by MIN function and MAX function, represents the most reliable method for non-overnight time comparisons.

To ensure maximum accuracy and ease of maintenance when performing these checks, several best practices should be followed:

Data Formatting: Always verify that all cells involved (C2, A2, B2) are consistently formatted as Time or a custom format that displays time. Inconsistent formatting can lead to comparing text strings or incorrect serial numbers.
Inclusion/Exclusion: Clearly define whether the time boundaries are inclusive (`>=`, ``, `<`). This decision should align precisely with the operational requirements of the analysis.
Overnight Handling: Recognize the limitations of the MIN/MAX structure. If the dataset includes ranges that cross midnight, a more complex logical structure involving the OR function must be implemented to correctly interpret the time fractions.
Custom Outputs: Leverage the flexibility of the IF function to return specific values, text, or calculations relevant to the business context, rather than being limited to simple Boolean text outputs like "Yes" and "No."

By applying this formula and adhering to these best practices, users can confidently automate the validation of event timing across large and complex datasets within Excel.

Additional Resources

The following tutorials explain how to perform other common operations in Excel, building upon the foundational knowledge of conditional logic and time management: