

# Excel: Check if Month and Year Match Between Dates

Authored by  
**Mohammed loot**

November 10, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Excel: Check if Month and Year Match Between Dates*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=15288>

## Mastering Date Comparisons in Excel: The Core Challenge

Working extensively with time-series data in [Microsoft Excel](#) frequently requires sophisticated comparison techniques. A particularly common and critical requirement in data analysis is the ability to compare two distinct dates while deliberately disregarding the specific day component. Analysts often need to determine solely if the month and year of two dates are identical, which is crucial for tasks such as accurately grouping financial data, verifying billing or subscription cycles, or performing high-level chronological reconciliation where daily granularity is unnecessary. The challenge lies in isolating these specific temporal elements from the complex date serial numbers that [Excel](#) uses internally.

Fortunately, [Excel](#) provides a highly efficient and concise formulaic solution. This method harnesses powerful, built-in date functions alongside textual [concatenation](#) to achieve this precise, component-based comparison. By transforming the date values into a single text string--composed uniquely of the month followed immediately by the year--we can create an objective basis for evaluating equality based exclusively on the relevant temporal criteria.

To implement this check efficiently, determining whether the month and year match between any two dates, we employ a structure that leverages the **MONTH** and **YEAR** functions to extract the numerical parts. The ampersand symbol (&) is then utilized as the [concatenation](#) operator, combining these numerical outputs into a single, comparable text value. This results in a clean, logical test that returns an undeniable result.

**=MONTH(A2)&YEAR(A2)=MONTH(B2)&YEAR(B2)**

This core formula is fundamentally designed to compare the combined month and year components extracted from the date housed in cell **A2** against the equivalent components derived from the date in cell **B2**. The outcome of this calculation is always a definitive [Boolean logic](#) value, returning **TRUE** if both dates share the exact same month and year combination, or **FALSE** immediately if they differ in either temporal component. This binary output is highly useful for subsequent conditional processing.

### Practical Application: Generating a Binary Output (TRUE/FALSE)

While understanding the underlying formula structure is essential, observing its practical application provides immediate clarity regarding its utility in real-world data analysis environments. Consider a common scenario in business intelligence where an analyst must compare two sets of dates--for instance, verifying the alignment between a list of initial order placement dates and corresponding shipment dates. A swift and accurate method is needed to confirm temporal alignment based solely on the month and year, which is crucial for financial auditing, inventory

verification, or supply chain reconciliation where the specific day of the transaction holds less importance than the billing period.

For the purpose of this demonstration, we will assume a sample dataset organized within an [Excel](#) worksheet. Column A holds the primary set of dates (e.g., Order Date), and Column B contains the secondary set intended for comparison (e.g., Shipment Date):

	A	B	C	D	E
1	<b>Date List 1</b>	<b>Date List 2</b>			
2	1/1/2023	1/4/2023			
3	1/15/2023	1/25/2022			
4	2/4/2023	2/14/2023			
5	3/22/2023	4/15/2023			
6	4/15/2023	12/1/2022			
7	5/19/2023	5/16/2023			
8	6/1/2023	6/1/2020			
9	8/19/2023	8/17/2023			
10	9/7/2023	10/30/2023			
11					
12					
13					
14					
15					
16					
17					

To initiate the comparison process, we start by selecting cell **C2**, which will serve as the output column for our results. We carefully input the core comparison formula to evaluate the dates presented in cells **A2** and **B2**. It is vital to ensure that the cell references are correctly positioned for the first row of data before proceeding:

**=MONTH(A2)&YEAR(A2)=MONTH(B2)&YEAR(B2)**

After successfully entering the formula into **C2**, analysts can leverage Excel's powerful autofill functionality. By simply clicking and dragging the formula handle (the small square in the lower-right corner of C2) downwards, this sophisticated comparison logic is instantaneously applied across all corresponding rows in Column C. This action systematically evaluates every pair of dates in Columns A and B against the criterion of matching month and year. The resulting output in Column C will clearly display a **TRUE** or **FALSE** value for each transaction, offering instant

verification of temporal congruence across the entire dataset.

	A	B	C	D	E	F
1	<b>Date List 1</b>	<b>Date List 2</b>	<b>Month &amp; Year Match?</b>			
2	1/1/2023	1/4/2023	TRUE			
3	1/15/2023	1/25/2022	FALSE			
4	2/4/2023	2/14/2023	TRUE			
5	3/22/2023	4/15/2023	FALSE			
6	4/15/2023	12/1/2022	FALSE			
7	5/19/2023	5/16/2023	TRUE			
8	6/1/2023	6/1/2020	FALSE			
9	8/19/2023	8/17/2023	TRUE			
10	9/7/2023	10/30/2023	FALSE			
11						
12						
13						
14						
15						
16						
17						

As clearly demonstrated in the image above, Column C functions as the definitive indicator of the match status for each pair. A result of **TRUE** confirms that the dates in the corresponding cells in Columns A and B share an identical month and year combination. Conversely, a result of **FALSE** immediately signals a discrepancy in either the month, the year, or both. This fundamental approach, rooted in [Boolean logic](#), remains the most streamlined and efficient way to conduct large-scale date comparisons when a simple binary answer is required for subsequent data filtering, sorting, or conditional formatting rules.

## Enhancing Readability with Custom Text Output (The IF Function)

While the **TRUE** and **FALSE** outputs generated by the comparison formula are mathematically precise and ideal for internal conditional processing, they often lack the clarity required for reports or dashboards intended for non-technical stakeholders. To significantly enhance readability and ensure immediate comprehension of the results, the recommended best practice is to enclose the core comparison formula within the versatile [IF function](#). This integration allows us to return customized, descriptive text strings instead of the standard binary Boolean values.

By nesting our month and year comparison inside the [IF function](#), we must define its three

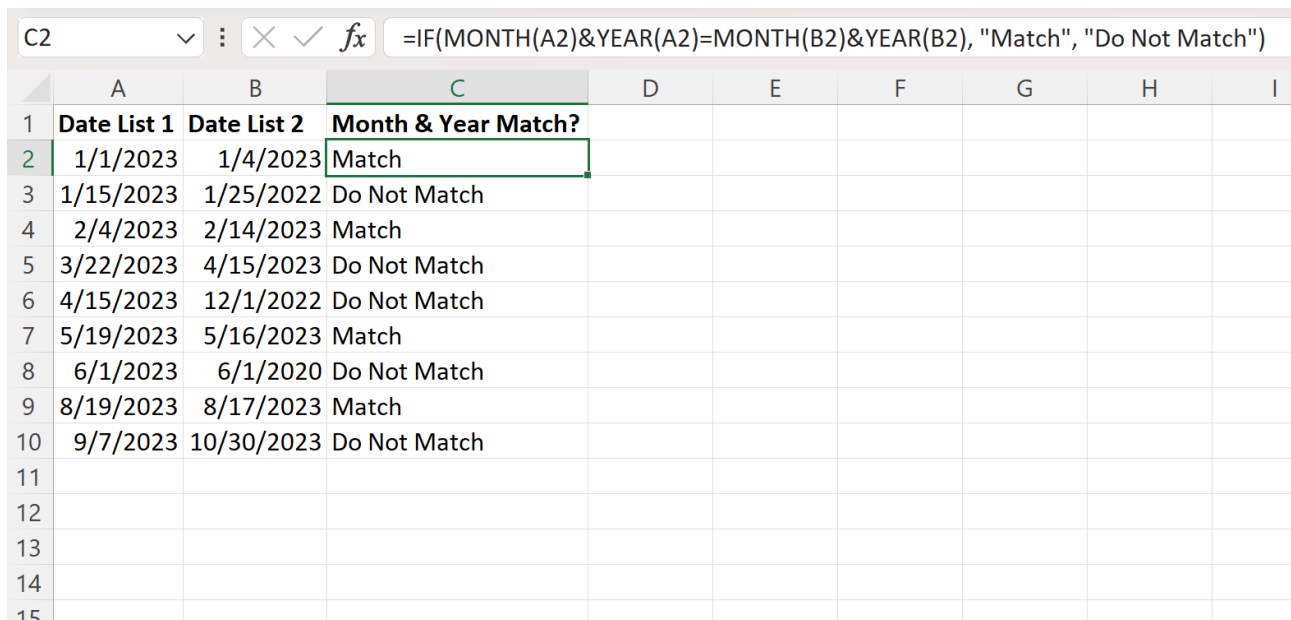
essential arguments clearly: first, the logical test (which is our original month/year equality check); second, the value to be returned if the test evaluates to **TRUE**; and third, the value to be returned if the test evaluates to **FALSE**. This structural modification elevates the usability of the dataset, making the results instantly accessible and unambiguous for any user reviewing the worksheet.

The revised formula structure, specifically engineered to output descriptive text strings such as "Match" or "Do Not Match" based on the outcome of the date comparison, dramatically improves data interpretation:

**=IF(MONTH(A2)&YEAR(A2)=MONTH(B2)&YEAR(B2), "Match", "Do Not Match")**

This powerful enhancement transforms the output from a technical verification into an easily digestible status report. When this descriptive formula is applied across the corresponding data range, the result instantly communicates the status of the date comparison, eliminating the need for users to interpret a raw [Boolean logic](#) outcome. This approach is invaluable when results are summarized, exported, or presented to management teams who require quick, actionable insights.

The screenshot below clearly illustrates the practical outcome achieved when this descriptive formula is applied throughout Column C, replacing the simple TRUE/FALSE format with meaningful text:



	A	B	C	D	E	F	G	H	I
1	Date List 1	Date List 2	Month & Year Match?						
2	1/1/2023	1/4/2023	Match						
3	1/15/2023	1/25/2022	Do Not Match						
4	2/4/2023	2/14/2023	Match						
5	3/22/2023	4/15/2023	Do Not Match						
6	4/15/2023	12/1/2022	Do Not Match						
7	5/19/2023	5/16/2023	Match						
8	6/1/2023	6/1/2020	Do Not Match						
9	8/19/2023	8/17/2023	Match						
10	9/7/2023	10/30/2023	Do Not Match						
11									
12									
13									
14									
15									

Notice the clarity provided by Column C, which now explicitly returns either **Match** or **Do Not Match**. This technique is strongly recommended for use in collaborative or shared spreadsheet environments, or in any scenario where the results must be rapidly filtered or aggregated based on easy-to-understand descriptive criteria, offering superior clarity compared to relying on standard

Boolean outputs alone.

## The Mechanics: Understanding MONTH, YEAR, and Concatenation

Achieving true mastery of this technique requires a fundamental understanding of the underlying mechanism that permits [Excel](#) to compare dates by their components rather than relying on the full, complex underlying date serial numbers. The efficiency, robustness, and reliability of this formula hinge on the precise, sequential execution of specific date functions combined with the vital [concatenation](#) operator.

Let us revisit the core comparison logic that orchestrates the entire process:

**=MONTH(A2)&YEAR(A2)=MONTH(B2)&YEAR(B2)**

The formula executes the comparison in three distinct, yet near-instantaneous, logical steps:

**Step 1: Component Isolation via Date Functions:** The initial and most critical task involves using the [MONTH function](#) and the [YEAR function](#). The [MONTH function](#) extracts the month number (an integer ranging from 1 to 12) from the date serial number in the specified cell (e.g., **A2**). Concurrently, the [YEAR function](#) pulls out the corresponding four-digit year. By applying these functions, we successfully isolate the specific numerical values necessary for the comparison, purposefully discarding the day value which would otherwise introduce unwanted variability into the equality check.

**Step 2: Combining Elements through Concatenation:** The ampersand symbol (&) serves as the textual joining mechanism--the [concatenation](#) operator--in [Excel](#). It is employed here to join the numerical output of the [MONTH function](#) and the [YEAR function](#) into a single, cohesive text string. For example, if cell **A2** contains the date 1/15/2023, the calculation **MONTH(A2)&YEAR(A2)** evaluates to 1 & 2023, resulting in the text string **12023**. This conversion into a single text unit is essential as it forces Excel to perform a simple textual equality comparison, effectively neutralizing potential numerical ambiguity.

**Step 3: Logical Comparison:** The final stage uses the equals sign (=) to perform a logical comparison between the two resulting concatenated text strings (one from **A2** and one from **B2**). If both strings are identical--for instance, **12023** equals **12023**--the formula outputs **TRUE**. If the strings differ in any way (e.g., **12023** vs. **22023**), the result is **FALSE**. This powerful, yet simple, logic ensures the comparison is based purely and reliably on the required month and year combination, successfully bypassing the day component.

This methodology provides a robust solution, allowing analysts to sidestep the inherent complexity of comparing full Excel date serial numbers, focusing instead only on the specific temporal

components critical for their data analysis needs.

## Summary and Alternative Approaches for Date Manipulation

The technique outlined throughout this guide, which strategically combines the **MONTH** function, the **YEAR** function, and the [concatenation](#) operator, represents the most straightforward and highly efficient approach for verifying month and year equality in [Excel](#). Whether the analytical requirement calls for a binary (TRUE/FALSE) result suitable for conditional logic or a highly customized text output achieved by nesting the logic within the [IF function](#), this formula structure offers remarkable adaptability and forms a cornerstone of advanced date manipulation skills.

While the concatenation method is widely favored for its inherent simplicity and guaranteed reliability, it is important for expert users to be aware of alternative techniques. For example, the **TEXT** function offers another path to achieve a similar result by directly formatting the date into a comparable string format. An example of this alternative syntax would be: `=TEXT(A2, "yyyymm")=TEXT(B2, "yyyymm")`. However, the combination of the `MONTH` and `YEAR` functions often retains preference among seasoned analysts because it is less susceptible to potential regional formatting complexities that can occasionally arise and complicate the output when using the `TEXT` function across diverse international datasets.

Mastering these core date manipulation techniques is indispensable for any analyst who frequently processes time-series data, performs data aggregation, or needs to summarize large volumes of information based on defined monthly or annual periods. These foundational formulas ensure robust data governance, enable precise temporal checks, and ultimately facilitate the creation of more accurate and insightful business reports.

## Additional Resources for Excel Mastery

We highly recommend delving into the following tutorials and official resources to further expand your expertise in advanced functions and critical data manipulation tasks within [Excel](#):

Advanced [IF function](#) Nesting Techniques

Comprehensive Guide to [MONTH](#) and [YEAR](#) Functions

Best Practices for Handling Date and Time Data in Spreadsheets