

Extracting Minutes from Datetime in Excel: A Step-by-Step Guide

Authored by
Mohammed loot

November 11, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Extracting Minutes from Datetime in Excel: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16749>

Introduction to Time Extraction and the MINUTE Function

The core requirement for effective data analysis in spreadsheets often hinges on the ability to accurately segment and manipulate time-based information. When confronted with large data sets that include combined date and time stamps--commonly referred to as [Datetime](#) values--analysts frequently need to isolate specific temporal components, such as the hour, minute, or second, for aggregation, sorting, or complex calculations. For example, if you are tasked with analyzing network logs or tracking e-commerce transaction timing, determining the distribution of events that occurred precisely during the 30th minute of every hour, independent of the specific day or year, becomes a critical analytical step. Fortunately, [Excel](#) provides a powerful and dedicated suite of functions designed to handle these precise time extraction tasks both efficiently and reliably.

Central to this suite of tools is the specialized [MINUTE function](#). This utility serves as a powerful means of isolating and retrieving the minute component from any valid time entry or [Datetime](#) stamp. The function processes the temporal data and returns a straightforward integer value ranging from 0 to 59, representing the minute past the hour. Mastering the deployment of this simple, yet crucial function is essential for anyone involved in time series analysis or requiring fine-grained control over temporal data within the [Excel](#) environment. This comprehensive guide will explore the mechanics and practical applications of the [MINUTE function](#), guiding you through the successful extraction of minutes from complicated date and time fields.

A significant advantage of utilizing the built-in [MINUTE function](#), rather than relying on complex string manipulation or custom formula logic, lies in its direct interaction with [Excel](#)'s inherent time serial number system. This foundational connection guarantees that the extraction process is always accurate, irrespective of how the cell is visually formatted on the screen. Whether the source cell displays a short time, a long time, or a comprehensive date and time stamp, the function consistently targets the underlying fractional time portion of the serial number, thus assuring a precise and reliable extraction of the required minute value for subsequent calculations or detailed reporting.

The Syntax and Mechanics of the MINUTE Function

The syntax required to deploy the [MINUTE function](#) is exceptionally clear and concise: `=MINUTE(serial_number)`. The required argument, `serial_number`, refers to the [Datetime](#) value from which the minutes must be extracted. This argument is highly flexible; it can be a direct cell reference containing the date and time, a formula that returns a time value (such as `NOW()` or `TIMEVALUE()`), or the raw serial number itself. Upon processing the input, the function reliably outputs an integer, representing the minutes that have passed since the last hour, with the range strictly limited between zero (0) and fifty-nine (59).

To illustrate the fundamental application, consider a practical scenario where a comprehensive

[Datetime](#) stamp, such as 11/25/2023 14:45:30, has been entered into cell **A2** of your worksheet. If your primary goal is simply to retrieve the '45' minute component, the process is straightforward: you reference the cell directly within the function structure. The resulting output is a clean, stand-alone integer (45), which is immediately available for any necessary numerical use, such as sorting or mathematical operations. Utilizing this direct cell reference method is the fastest and most dependable way to isolate this specific temporal unit.

For instance, assuming cell **A2** contains the full timestamp value, the formula used to extract only the minutes is written as follows, clearly directing [Excel](#) to the source data:

=MINUTE(A2)

This command explicitly instructs [Excel](#) to analyze the numerical serial value resident in **A2** and return solely the minutes component. It is absolutely crucial to verify that the source cell **A2** is correctly formatted as a numerical [Datetime](#) entry, or at minimum, contains a time value recognized by the system. Failure to use valid numerical time data will likely cause the function to return an error (such as #VALUE!) or an incorrect result (such as 0 if only a date without any time component is present).

Practical Step-by-Step Guide: Extracting Minutes from Large Datasets

To demonstrate the full practical application of the [MINUTE function](#) across an extensive list of records, let us consider a typical scenario involving a sales transaction log. Imagine you have a worksheet logging sales data, where Column A meticulously records the exact date and time of each sale, while Column B tracks the quantity sold. Your objective is to efficiently populate a new column that isolates only the exact minute of the transaction. This isolated minute value can then be readily used to categorize sales into custom time bins (e.g., 5-minute intervals) or to precisely determine the peak minute activity periods throughout the day.

We begin with the sample dataset shown below, where Column A clearly displays the combined date and time stamps for each transaction record:

	A	B	C	D	E
1	Datetime	Sales			
2	1/1/2023 4:14:59	14			
3	1/2/2023 10:15:01	19			
4	1/2/2023 8:14:09	22			
5	1/2/2023 23:34:45	24			
6	1/3/2023 12:14:50	38			
7	1/4/2023 1:15:24	10			
8	1/5/2023 5:27:48	14			
9	1/5/2023 5:38:35	13			
10	1/6/2023 16:45:58	18			
11					
12					
13					
14					
15					

Our goal is to populate a new column, specifically Column C, with only the minute value derived from the corresponding [Datetime](#) stamp found in Column A. To initiate this process, select cell **C2**, which is the starting point for our new extracted data. We then input the specific formula that targets the first timestamp located in cell **A2**. This action clearly signals to [Excel](#) precisely which underlying serial number it must analyze to extract the required minute value.

The required formula to be entered into cell **C2** is as follows:

=MINUTE(A2)

After entering the formula into **C2**, press the Enter key to calculate the result for the first row. The next, and most efficient, step is to apply this formula consistently down the entire dataset. This is accomplished using the **fill handle**--the small square located at the bottom-right corner of the selected cell **C2**. By clicking and dragging this handle downward, the relative cell reference (A2) automatically adjusts for each subsequent row (becoming A3, A4, A5, and so on). This mechanism ensures that the minute extraction is performed accurately for every single transaction timestamp within the column.

Once the drag operation is complete, Column C will instantly populate with the newly extracted minute values, successfully isolating the temporal component from the complex original [Datetime](#) stamp. This result, clearly demonstrated in the resulting table below, provides a clean numerical

column that is perfectly suited for subsequent detailed analysis, pivoting, aggregation, or the creation of custom time bins.

	A	B	C	D	E
1	Datetime	Sales	Minutes		
2	1/1/2023 4:14:59	14	14		
3	1/2/2023 10:15:01	19	15		
4	1/2/2023 8:14:09	22	14		
5	1/2/2023 23:34:45	24	34		
6	1/3/2023 12:14:50	38	14		
7	1/4/2023 1:15:24	10	15		
8	1/5/2023 5:27:48	14	27		
9	1/5/2023 5:38:35	13	38		
10	1/6/2023 16:45:58	18	45		
11					
12					
13					
14					

Advanced Techniques: Combining Minutes and Seconds (MM:SS)

While simple minute extraction is often sufficient, many data reporting requirements necessitate combining different time components into a customized, specific format, such as displaying minutes and seconds together in the standard MM:SS format. Achieving this combination requires integrating the [MINUTE function](#) with the `SECOND` function, alongside careful use of [Concatenation](#) operators and specific number formatting to maintain visual consistency.

To perform this advanced, combined extraction, we utilize the ampersand symbol (&) as the primary [Concatenation](#) tool, inserting a colon (":") as a fixed delimiter between the two time components. The formula starts logically with the [MINUTE function](#) to retrieve the first component. However, a potential issue arises with the seconds value: the `SECOND` function returns a simple integer. If the seconds value is less than 10 (e.g., 5 seconds), it will return the numeral '5', which violates the standard MM:SS format that requires a leading zero (e.g., 05).

To successfully resolve this critical formatting inconsistency and ensure that all single-digit seconds are correctly displayed as two digits (e.g., 05 instead of 5), we must wrap the `SECOND` function within the highly versatile [TEXT function](#). The [TEXT function](#) allows us to specify the exact

output format using "00" as the format code, which automatically guarantees a two-digit result, padded with a leading zero if necessary. This powerful combination results in a robust and reliable method for presenting time components in a standardized string format suitable for display purposes.

The complete formula required to extract and combine the minutes and seconds from a [Datetime](#) located in cell **A2**, ensuring the output is formatted as MM:SS, is as follows:

=MINUTE(A2)&":"&TEXT(SECOND(A2), "00")

This sophisticated formula ensures that the final output is a formatted text string rather than a numerical time value, a result often preferred when the data is purely intended for display or detailed reporting. The visual example below confirms how this combined function successfully extracts and formats both the minute and second components, even when the original timestamp is highly detailed and complex:

	A	B	C	D	E
1	Datetime	Sales	Minutes	Minutes & Seconds	
2	1/1/2023 4:14:59	14	14	14:59	
3	1/2/2023 10:15:01	19	15	15:01	
4	1/2/2023 8:14:09	22	14	14:09	
5	1/2/2023 23:34:45	24	34	34:45	
6	1/3/2023 12:14:50	38	14	14:50	
7	1/4/2023 1:15:24	10	15	15:24	
8	1/5/2023 5:27:48	14	27	27:48	
9	1/5/2023 5:38:35	13	38	38:35	
10	1/6/2023 16:45:58	18	45	45:58	
11					
12					
13					
14					

Deciphering Excel's Time System: The Datetime Serial Number

To achieve true mastery over time extraction in [Excel](#), it is necessary to understand the underlying architecture that governs date and time values: the serial number system. Unlike handling dates and times as textual strings, [Excel](#) stores every date and time as a numerical value. Dates are represented by the integer portion of this serial number (counting the number of days elapsed

since January 1, 1900), while time is represented by the crucial fractional portion of that number.

This fractional component is the key to understanding how the [MINUTE function](#) operates. Given that there are 1,440 total minutes in a full day (24 hours multiplied by 60 minutes), the time component of the serial number is calculated as a precise fraction of 1. For instance, 12:00 PM (noon) is internally stored as 0.5, because it marks exactly halfway through the day. The [MINUTE function](#) intelligently interprets this fractional value, calculates the total minutes from midnight, and then isolates the remainder after division by 60, thereby correctly extracting the minutes component of the current hour.

Understanding this numerical mechanism clarifies why the visual formatting of the source cell has absolutely no bearing on the calculation's accuracy. Even if a cell is formatted to display only the date (e.g., 11/25/2023), if the underlying serial number contains a time component (e.g., 45242.615, where .615 represents the time portion), the [MINUTE function](#) will accurately read that fractional value and return the corresponding minutes. Conversely, if a cell contains only a date with no explicit time (meaning the fractional part is zero), the [MINUTE function](#) will correctly return 0, as there are zero minutes past midnight.

Troubleshooting Common Errors and Unexpected Results

Although the [MINUTE function](#) is highly reliable, users frequently encounter errors or unexpected outcomes, which usually originate from incorrect data types or subtle formatting conflicts. Addressing these common pitfalls ensures smooth data processing and accurate analytical results. The most frequently seen error is the `#VALUE!` error. This typically occurs when the input argument is not a recognized numeric date or time serial number, but instead a textual string that [Excel](#) cannot successfully coerce into a numerical format.

To effectively troubleshoot a `#VALUE!` error, the first step is to verify that the referenced source cell genuinely contains a numerical [Datetime](#) entry. If the data was imported or manually entered as a text string (which often happens when regional date settings clash with the input format), you may need to employ a conversion function such as `DATEVALUE` or `TIMEVALUE` to transform the text into a proper serial number before supplying it as the argument to the [MINUTE function](#). Alternatively, applying the `VALUE` function directly to the source cell can sometimes prompt [Excel](#) to force the text recognition into a number.

Another common confusion arises when the function returns a result of 0 despite the user expecting a specific minute value (e.g., 30). This usually indicates that the source cell contains only a date and lacks an explicit time component, causing the time to default to 12:00:00 AM, or 0 in the fractional serial number. If your analytical objective is to extract minutes, you must confirm that your source data includes the time stamp. If the data indeed only contains a date, the [MINUTE function](#) is technically operating correctly by returning 0. Finally, pay attention to the resulting cell's

format: since the [MINUTE function](#) returns an integer (0-59), the cell displaying the result should be formatted as **General** or **Number**. If it is inadvertently set to a Time format, the numerical result (e.g., 45) will be displayed incorrectly as a time value based on the serial number system.

Conclusion and Expanding Your Temporal Analysis Toolkit

The [MINUTE function](#) stands as an indispensable tool for data professionals and analysts who routinely manipulate temporal data within [Excel](#). Its remarkable simplicity of execution, combined with the inherent accuracy provided by [Excel](#)'s reliable serial number system, establishes it as the definitive method for isolating the minute component from any sophisticated [Datetime](#) stamp. Whether utilized as a standalone function for quick extraction or integrated into more complex formulas involving [Concatenation](#) and stringent formatting via the [TEXT function](#), it delivers the necessary granularity for detailed, time-based reporting and advanced analytical tasks.

Crucially, the principles and logic demonstrated throughout this guide extend directly to all other temporal extraction functions in [Excel](#), enabling comprehensive control over every aspect of time data. The companion functions operate on the exact same underlying logic, targeting their respective units within the fractional serial number. These related functions should be considered the next step in building your time analysis proficiency:

hour(serial_number): Extracts the hour component, returning an integer between 0 and 23.

second(serial_number): Extracts the seconds component, returning an integer between 0 and 59.

timevalue(time_text): Converts a time string that is recognized as text into a numerical [Excel](#) time serial number.

By fully mastering the application of the [MINUTE function](#), users acquire a fundamental skill that can be effortlessly scaled to address a wide range of analytical challenges, ensuring that temporal data is always exploited to its maximum potential within the spreadsheet environment.

Additional Resources for Data Proficiency

For users seeking to further expand their proficiency in temporal and broader data manipulation techniques within [Excel](#), the following tutorials and resources provide detailed explanations on closely related operations:

How to combine date and time values into a single, valid Datetime stamp.

Methods for accurately calculating the elapsed time or difference between two specific Datetime stamps.

Advanced uses of the [TEXT function](#) for complex and custom number formatting beyond time components.