

# Average Every Nth Row in Excel (With Example)

Authored by  
**stats writer**

November 17, 2025

## RECOMMENDED CITATION

stats writer (2025). *Average Every Nth Row in Excel (With Example)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95035>

Calculating the average of every Nth row in Excel is a powerful and specialized technique essential for robust data analysis. This methodology moves beyond simple overall averages, allowing analysts to extract specific insights by systematically sampling data at fixed intervals. Such methods are particularly vital when dealing with large time-series datasets or detailed transactional logs where examining every single point can obscure underlying patterns.

This approach is fundamentally used to generate a form of rolling average or sampled summary statistic. When applied to fields like finance, epidemiology, or quality control, averaging every Nth observation helps in dampening short-term volatility, thereby highlighting long-term or cyclical trends. For instance, calculating the average of a metric every 7th day can reveal weekly cycles that might be invisible when examining daily fluctuations. The resulting data stream provides a smoother curve, minimizing the disruptive effect of "noise" or random variation inherent in high-frequency data.

Furthermore, isolating specific data points using the Nth row technique is highly effective for outlier detection. When a single data point deviates significantly from the calculated periodic average, it becomes a strong candidate for being an outlier, prompting further investigation. Understanding this technique is crucial for any user looking to deepen their command of Excel's analytical capabilities and move beyond basic statistical functions.

## The Core Formula: Using Array Logic

To successfully calculate the average of every Nth row dynamically in Excel, we must employ a specific combination of functions, typically entered as an array formula (though modern versions of Excel with dynamic arrays may handle this without manual CSE entry). This complex formula relies on using the remainder of row division to filter the data precisely. Below is the general structure:

The fundamental formula used to average every Nth row within a defined range in Excel is:

**=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21))),n)=0,A2:A21))**

This powerful construction calculates the average only of those values within the specified range (here, **A2:A21**) that meet the condition defined by the value of **n**. It is essential to understand the role of each component within this sophisticated calculation, which hinges on logical filtering.

To customize this calculation for specific intervals, you only need to adjust two parameters: the data range (**A2:A21**) and the interval value (**n**). For example, to calculate the average of every **4th** row, we replace **n** with 4:

**=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21))),4)=0,A2:A21))**

The following detailed examples illustrate how to implement this formula and verify the results manually across different interval values.

## Deconstructing the Formula Components

Understanding how the nested functions interact is key to customizing this solution. The formula works from the inside out to generate an array of TRUE/FALSE values corresponding to the rows we wish to include in the average. The process involves three main function families: ROW, MOD, and IF.

**ROW(Range):** This component generates an array of absolute row numbers for the specified range.

**MIN(ROW(Range)):** This determines the starting row number of the range. Subtracting this value from the absolute row numbers effectively normalizes the row count, resulting in a zero-based index for the range: {0; 1; 2; ...}. This normalization is critical because it ensures the counting of Nth rows starts correctly from the first cell in the chosen range, not from row 1 of the spreadsheet.

**MOD(..., n):** The MOD function calculates the remainder after division. By testing if the remainder of the normalized row number divided by **n** is equal to zero, we isolate every Nth row. If the remainder is zero, the row index is perfectly divisible by N (meaning it is an Nth row).

**IF(MOD(...) = 0, Range):** The IF statement checks the remainder condition. If TRUE (remainder is 0), it returns the corresponding value from the range; if FALSE, it returns the value FALSE, which the outer AVERAGE function ignores.

## Example: Average Every Nth Row in Excel

### Practical Application: Averaging Every 4th Row (N=4)

To demonstrate the efficacy of this array formula, consider a dataset representing 20 sequential measurements stored in column A, starting from cell A2 and ending at A21. Our objective is to calculate the average value based on a sampling rate of every fourth measurement (N=4). This might be used, for example, to analyze quarterly metrics derived from monthly data points.

The dataset used for this example is presented below. Note that the range spans from row 2 through row 21, totaling twenty data entries:

	A	B	C	D	E	F
1	<b>Values</b>					
2	4					
3	8					
4	12					
5	12					
6	10					
7	14					
8	15					
9	19					
10	17					
11	13					
12	22					
13	25					
14	29					
15	24					
16	25					
17	24					
18	30					
19	24					
20	25					
21	30					

To isolate and average the values in every 4th row within this specific data range (A2:A21), we input the interval value of 4 into the primary formula structure. Remember that, depending on your version of Excel, you may need to enter this using **Ctrl+Shift+Enter** to confirm it as an array formula.

The required formula is:

**=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21))),4)=0,A2:A21))**

Upon execution, the calculation proceeds by first normalizing the rows (creating indices 0, 1, 2, 3, 4, etc.) and then selecting only those rows where the index is perfectly divisible by 4 (i.e., indices 0, 4, 8, 12, 16). The result of this calculation, as shown in the subsequent demonstration, provides the mean of the sampled data points.

### Verifying the Calculated Result (N=4)

The screenshot below illustrates the formula entered into the designated cell, yielding the final average. This demonstrates the seamless operation of the nested array logic, delivering the

statistical summary without requiring any manual data filtering or selection.

C2		=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21)),4)=0,A2:A21))							
	A	B	C	D	E	F	G	H	
1	Values		Average of Every 4th Row						
2	4		18						
3	8								
4	12								
5	12								
6	10								
7	14								
8	15								
9	19								
10	17								
11	13								
12	22								
13	25								
14	29								
15	24								
16	25								
17	24								
18	30								
19	24								
20	25								
21	30								
22									

As confirmed by the formula output, the calculated average of every 4th row in the range **A2:A21** is precisely **18**. It is good practice in data analysis to manually verify the first calculation to ensure the formula logic aligns with expectations, especially when dealing with complex array filtering.

### Manual Verification for N=4

We can confirm the accuracy of the result by manually identifying the values that correspond to the 4th interval. Since our normalized row count begins at 0 (cell A2), the selected values correspond to the normalized indices 0, 4, 8, 12, and 16, which map directly to the actual rows A2, A6, A10, A14, and A18 in the spreadsheet.

	A	B	C	D
1	<b>Values</b>		<b>Average of Every 4th Row</b>	
2	4		18	
3	8			
4	12			
5	12			
6	10			
7	14			
8	15			
9	19			
10	17			
11	13			
12	22			
13	25			
14	29			
15	24			
16	25			
17	24			
18	30			
19	24			
20	25			
21	30			
22				

The values identified by the formula are: 4 (A2), 10 (A6), 17 (A10), 29 (A14), and 30 (A18). The calculation proceeds as follows:

$$\text{Average} = (4 + 10 + 17 + 29 + 30) / 5 = 18$$

This manual calculation perfectly matches the result derived from the complex array formula, confirming the correct application of the MOD and ROW functions.

### Adjusting the Sampling Interval (N=6)

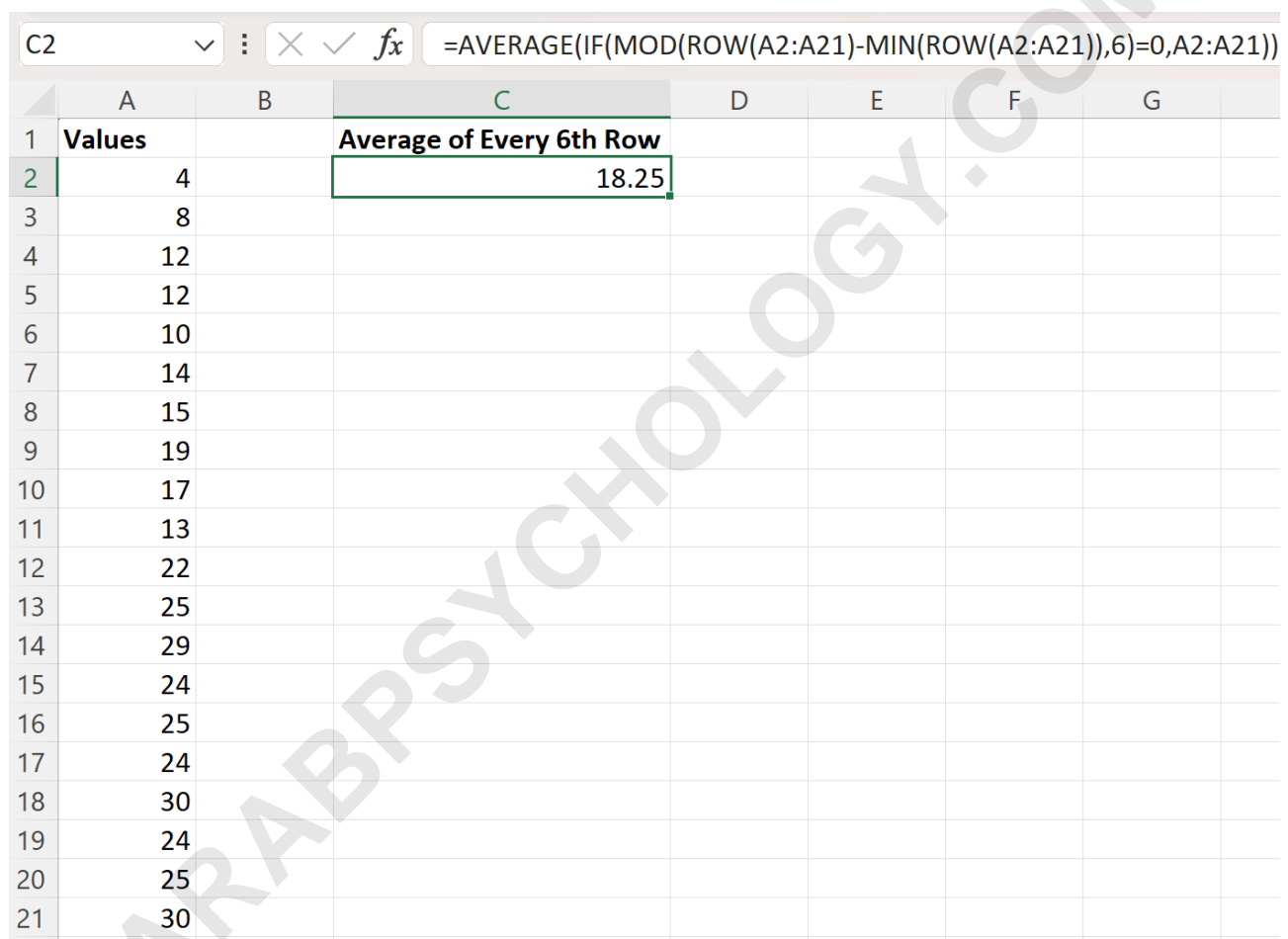
The utility of this formula lies in its flexibility. By simply modifying the parameter **n**, we can instantly recalculate the average based on a different sampling interval without altering the dataset or the range definition. This capability is invaluable for sensitivity testing or when comparing different cyclic periods within the same data stream.

Suppose we now wish to calculate the average of every **6th** row in the same range, **A2:A21**. We change the divisor in the MOD function from 4 to 6. This modification immediately recalibrates the filtering logic to select rows where the normalized index is divisible by six (i.e., indices 0, 6, 12, 18).

The revised formula targeting the 6th interval is:

**=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21))),6)=0,A2:A21))**

When implementing this change in the spreadsheet, the result reflects the new, wider sampling interval. The screenshot below shows the revised output when N=6 is used. Notice the difference in the calculated mean compared to the N=4 example, demonstrating how sampling density impacts the resulting summary statistic.



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G
1	Values		Average of Every 6th Row				
2	4		18.25				
3	8						
4	12						
5	12						
6	10						
7	14						
8	15						
9	19						
10	17						
11	13						
12	22						
13	25						
14	29						
15	24						
16	25						
17	24						
18	30						
19	24						
20	25						
21	30						

The computed output indicates that the average of every 6th row within the specified range **A2:A21** is **18.25**. This figure is based on a smaller number of data points compared to the N=4 calculation, as fewer values satisfy the divisibility criterion when the divisor increases.

### Manual Verification for N=6

To confirm this result, we identify the data points that correspond to the normalized indices 0, 6, 12, and 18. These correspond to cells A2, A8, A14, and A20, respectively, within the data range.

The selected values are highlighted in the image below:

	A	B	C	D
1	<b>Values</b>		<b>Average of Every 6th Row</b>	
2	4		18.25	
3	8			
4	12			
5	12			
6	10			
7	14			
8	15			
9	19			
10	17			
11	13			
12	22			
13	25			
14	29			
15	24			
16	25			
17	24			
18	30			
19	24			
20	25			
21	30			
22				

The four sampled values are: 4 (A2), 15 (A8), 29 (A14), and 25 (A20). We calculate the average of these four numbers:

$$\text{Average} = (4 + 15 + 29 + 25) / 4 = \mathbf{18.25}$$

This verification confirms that the array formula correctly filters the data based on the normalized row indices and interval  $n$ , providing a reliable calculation of the sampled mean.

### Handling Different Starting Points (Offsetting the Count)

A crucial aspect of this sampling technique is that the current formula always begins its count from the very first cell in the defined range (due to the **MIN(ROW(...))** normalization, which sets the starting index to 0). However, in some statistical or chronological contexts, you might need to start sampling not on the 1st row, but on the 2nd or 3rd row, and then continue every Nth row thereafter.

To implement a starting offset, we introduce an adjustment parameter,  $k$ , into the MOD function. If

you want the sampling to start on the **k+1** row, you modify the core logic to look for a specific remainder, instead of checking for a remainder of 0. The revised condition becomes:

`=AVERAGE(IF(MOD(ROW(Range)-MIN(ROW(Range))), n) = k, Range))`

For example, to calculate every 4th row, starting from the **3rd** data point (where  $N=4$  and  $k=2$ , since the normalized row count is 0, 1, 2...), the formula would be: `=AVERAGE(IF(MOD(ROW(A2:A21)-MIN(ROW(A2:A21))), 4)=2, A2:A21))`. This adjustment provides absolute control over where the periodic sampling begins within the dataset, ensuring the technique meets precise data analysis requirements.

## Dynamic Range Management

While our examples utilize a fixed range (A2:A21), highly functional Excel models benefit from dynamic range definition. Using functions like **OFFSET** or **INDEX** in conjunction with **COUNTA** allows the calculation to automatically adjust as new data is added to the column. This removes the necessity of manually updating the range endpoints (A21 in our examples) every time the dataset grows.

A dynamic range implementation ensures that the Nth row average is always applied to the entirety of the relevant data column, maximizing efficiency and minimizing the risk of calculation errors due to outdated range references. For large-scale data logging operations, mastering dynamic range definitions is essential for reliable statistical output.