

How to Transpose Every N Rows in Google Sheets Easily

Authored by
stats writer

January 30, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Transpose Every N Rows in Google Sheets Easily*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128608>

Understanding how to manipulate data structures efficiently is fundamental when working with large datasets. In Google Sheets, transposing data involves the critical process of rearranging data by switching the orientation of rows and columns. While simple copying and pasting might suffice for minimal tables, this manual approach quickly becomes unsustainable and error-prone when dealing with significant volumes of information.

A more sophisticated and scalable solution exists for tackling complex transposition challenges, specifically when needing to transpose data every 'N' rows--a common requirement in data analysis and formatting. Instead of relying solely on the built-in TRANSPOSE function, which typically transposes entire ranges, we utilize a powerful combination of array manipulation functions. This technique leverages functions like INDEX function and ROW function to precisely control the data rearrangement, allowing you to specify the exact number of rows (N) to group before switching their orientation to columns. This article details the highly efficient method for segmenting and transposing data in a controlled manner.

Advanced Data Manipulation: Transposing Data Every N Rows

The core challenge in conditional transposition is dynamically selecting non-contiguous data segments (N rows) and arranging them horizontally. This specialized requirement necessitates a complex formula that combines multiple functions to calculate the precise row index needed for each new cell location. The following syntax provides the blueprint for transposing data from Column A based on an interval of 'N' rows:

=INDEX(\$A:\$A,ROW(A1)*5-5+COLUMN(A1))

This powerful formula, as written above, is specifically designed to transpose segments of data consisting of exactly five rows from the source column (Column A) into a new row structure. It achieves this by calculating the specific offset required to jump to the beginning of the next segment of data as it is dragged across and down the spreadsheet. Understanding the variables and how they interact is crucial for successful implementation.

Important Consideration: To adapt this method for different segment lengths, you only need to modify the two instances of the number **5** within the formula. These numbers dictate the size of the block (N) that will be transposed from a vertical orientation into a horizontal one. Always ensure both numbers are changed consistently to maintain mathematical integrity.

Deconstructing the N-Row Transposition Formula

To fully appreciate the efficacy of this method, it is beneficial to analyze the components of the formula. This calculation relies heavily on reference functions that return the current position of the

cell where the formula is placed, allowing the output to dynamically adjust as the formula is copied to new locations.

The formula structure is centered around the INDEX function: `INDEX($A:$A, row_number)`. The goal is to calculate the precise `row_number` in Column A that corresponds to the output position in the transposed table. The full row calculation logic is: `ROW(A1) * N - N + COLUMN(A1)`, where 'N' is 5 in the example provided.

The components work as follows:

`$A:$A`: This is the absolute reference to the entire source column containing the data we wish to transpose. Using absolute reference ensures the range remains fixed regardless of where the formula is copied.

`ROW(A1)`: This section determines the current row number of the output area. When the formula is in the first row of the output, `ROW(A1)` returns 1. When dragged down to the second row, it calculates `ROW(A2)`, which returns 2, and so on. This handles the vertical progression of the transposed blocks.

`* 5 - 5`: This is the core engine for segmenting. If $N=5$, $(ROW(A1) * 5)$ provides the starting point of the next block. Subtracting 5 (N) normalizes the starting index back to 1 for the first row of output, 6 for the second row, 11 for the third row, etc.

`+ COLUMN(A1)`: This component is crucial for horizontal expansion. The COLUMN function returns the column number (1 for Column A, 2 for Column B, etc.). As the formula is dragged horizontally, `COLUMN(A1)` becomes `COLUMN(B1)` (returning 2), `COLUMN(C1)` (returning 3), and so forth. This increment ensures that within each transposed block, the formula pulls the 1st, 2nd, 3rd, 4th, and 5th element from the original vertical list.

Practical Example: Setting Up the Dataset

To illustrate the efficiency and power of this array formula technique, let us walk through a practical scenario. Suppose we have a substantial list of data--in this case, 15 basketball team names--organized vertically within a single column in Google Sheets (Column A). Our objective is to restructure this linear list into a table where every five consecutive teams are grouped into their own distinct row.

The initial dataset, positioned in Column A, provides the vertical foundation for our transposition exercise. The goal is to convert the height of the data into width, segmenting the list at regular five-row intervals ($N=5$).

The following visual representation shows the starting point of our data manipulation task:

	A	B	C	
1	Mavs			
2	Warriors			
3	Hawks			
4	Kings			
5	Spurs			
6	Nets			
7	Magic			
8	Timberwolves			
9	Clippers			
10	Pacers			
11	Lakers			
12	Grizzlies			
13	Nuggets			
14	Raptors			
15	Knicks			
16				
17				

The task requires converting the entire dataset into blocks of five, meaning our final output will consist of three rows (15 total items / 5 items per row = 3 rows).

Applying the Formula (N=5)

Once the dataset is prepared in Column A, the next step involves initiating the transposition process by placing the formula in the intended starting cell of the new, transposed table. For this example, we will begin the output in cell **C2**. This separation allows us to easily compare the transposed output with the source data.

We input the formula designed for N=5 directly into cell **C2**:

=INDEX(\$A:\$A,ROW(A1)*5-5+COLUMN(A1))

When this formula is entered into **C2**, it calculates the index of the first element in the array, establishing the correct starting point for the transposition block based on the references A1/C2. The resulting view immediately after entering the formula demonstrates the extraction of the first element:

C1			$\text{=INDEX}(\$A:\$A, \text{ROW}(A1)*5-5+\text{COLUMN}(A1))$	
	A	B	C	D
1	Mavs		Mavs	
2	Warriors			
3	Hawks			
4	Kings			
5	Spurs			
6	Nets			
7	Magic			
8	Timberwolves			
9	Clippers			
10	Pacers			
11	Lakers			
12	Grizzlies			
13	Nuggets			
14	Raptors			
15	Knicks			
16				

Expanding the Results: Horizontal and Vertical Dragging

The true power of this formula manifests when it is copied across and down the spreadsheet. Because the formula relies on relative row and column references embedded within the ROW function and COLUMN function parts, dragging the cell dynamically updates the indices used to retrieve the source data.

First, we must complete the horizontal arrangement for the first block. Click on cell **C2** and drag the formula rightward until it occupies five total columns (C2 through G2). This horizontal expansion utilizes the `+ COLUMN(A1)` part of the formula, incrementing the index by one for each column, thereby retrieving elements 1 through 5 from Column A.

The successful horizontal drag will display the first five team names segmented from the original vertical list:

C1:G1 fx =INDEX(\$A:\$A,ROW(A1)*5-5+COLUMN(A1))

	A	B	C	D	E	F	G
1	Mavs		Mavs	Warriors	Hawks	Kings	Spurs
2	Warriors						
3	Hawks						
4	Kings						
5	Spurs						
6	Nets						
7	Magic						
8	Timberwolves						
9	Clippers						
10	Pacers						
11	Lakers						
12	Grizzlies						
13	Nuggets						
14	Raptors						
15	Knicks						
16							

Next, to process the subsequent blocks, click and drag the entire resulting row (C2:G2) downward. Drag the formula down until all 15 team names have been displayed. Dragging down activates the ROW(A1) component, which increases the overall index by 5 for each row, starting the next block segment correctly.

Interpreting the Transposed Output

The final outcome showcases the elegant solution provided by the combined array functions. The data has been successfully restructured from a single, long column into a compact, multi-row table, segmenting the input data based on the specified interval (N=5).

The results confirm the following structural transformation:

The first five team names located in the source Column A (A1 through A5) now constitute the data shown in the first row of the transposed output (C2:G2).

The subsequent segment of five team names (A6 through A10) forms the entirety of the second row of the output (C3:G3).

The final segment of five team names (A11 through A15) is correctly placed in the third row of the output (C4:G4).

The spreadsheet now displays the data in the required format:

C1:G3 fx `=INDEX($A:$A,ROW(A1)*5-5+COLUMN(A1))`

	A	B	C	D	E	F	G
1	Mavs		Mavs	Warriors	Hawks	Kings	Spurs
2	Warriors		Nets	Magic	Timberwolves	Clippers	Pacers
3	Hawks		Lakers	Grizzlies	Nuggets	Raptors	Knicks
4	Kings						
5	Spurs						
6	Nets						
7	Magic						
8	Timberwolves						
9	Clippers						
10	Pacers						
11	Lakers						
12	Grizzlies						
13	Nuggets						
14	Raptors						
15	Knicks						
16							

This organized presentation not only validates the formula's effectiveness but also dramatically improves readability and usability for further analysis or reporting purposes. This method is far superior to manual copying and pasting, especially for hundreds or thousands of rows.

	A	B	C	D	E	F	G
1	Mavs		Mavs	Warriors	Hawks	Kings	Spurs
2	Warriors		Nets	Magic	Timberwolves	Clippers	Pacers
3	Hawks		Lakers	Grizzlies	Nuggets	Raptors	Knicks
4	Kings						
5	Spurs						
6	Nets						
7	Magic						
8	Timberwolves						
9	Clippers						
10	Pacers						
11	Lakers						
12	Grizzlies						
13	Nuggets						
14	Raptors						
15	Knicks						
16							

Customizing the Transposition Interval

Flexibility is a key advantage of this method. While our example utilized N=5, the formula can be readily adapted to transpose data into blocks of any size N (e.g., every 3 rows, every 10 rows). The

customization requires only one simple modification: changing the numerical value that represents N within the segmentation logic.

To transpose a different multiple of rows, such as N=3, you must change both instances of the number 5 in the formula to 3. The revised structure would look like this:

=INDEX(\$A:\$A,ROW(A1)*3-3+COLUMN(A1))

When applying this modified formula, remember that the number of columns you drag the formula across must exactly match the new interval N. If N=3, you would drag the formula across three columns. If the number of columns does not match N, the resulting output will either leave gaps (if dragged too far) or fail to capture all data points within the block (if not dragged far enough).

Leveraging Array Formulas for Efficiency

The combination of the INDEX function, ROW function, and COLUMN function offers a highly robust methodology for conditional data transposing data within spreadsheets. This technique moves far beyond the limitations of simple copy-paste operations or the standard TRANSPOSE function, providing granular control necessary for sophisticated data restructuring.

Mastering such array manipulation formulas significantly enhances a user's capability to process and analyze complex data arrangements quickly and programmatically. This expertise is particularly valuable when handling raw data exports that often require consistent reformatting into predefined dimensional tables. By employing this method, users ensure data integrity and save considerable time previously spent on tedious manual adjustments.

The following tutorials explain how to perform other common tasks in Google Sheets: