How to Move Every Other Row to a New Column in Excel

Authored by

stats writer

February 11, 2026

The Importance of Efficient Data Structuring in Microsoft Excel

In the contemporary landscape of **data management**, the ability to manipulate and reorganize information within a **spreadsheet** is an essential skill for professionals across various industries. **Microsoft Excel** remains the industry standard for these tasks, offering a robust suite of tools designed to streamline **data analysis** and improve the clarity of information presentation. One common challenge users encounter is the need to rearrange datasets where information is stacked vertically, requiring every other row to be moved into a secondary column to facilitate side-by-side comparisons or more logical grouping.

This specific method of rearranging data involves a strategic approach to **cell references** and logical operations. By moving every other row to a new column, a user can transform a single-column list into a more structured multi-column format. This is particularly useful when dealing with exported data from external systems that may not be formatted for immediate analysis. Utilizing functions rather than manual **copy and paste** operations ensures **data integrity** and significantly reduces the risk of human error, which is a critical consideration in high-stakes reporting environments.

The process of reformatting a **spreadsheet** not only enhances the **data visualization** aspect of the project but also optimizes the document for further computational tasks. A well-organized layout allows for the application of advanced features like **PivotTables** or complex **lookup functions** more effectively. By mastering the formulas required to bifurcate data rows, users can improve their overall productivity and ensure that their **business intelligence** assets are both professional and highly functional.

Understanding the Logic of Alternate Row Manipulation

To move every other row to a new column in **Excel**, one must first understand the underlying **mathematical logic** used to identify specific rows. Every row in a worksheet is assigned a unique index number. By determining whether this index is an even or odd number, we can create a conditional statement that acts as a filter. This approach is far more scalable than manual selection, especially when dealing with thousands of data points that would be impossible to manage by hand.

The fundamental strategy relies on the **IF function**, which evaluates a specific condition and returns one value if the condition is true and another if it is false. When combined with functions that identify row properties, this becomes a powerful tool for **database management** within a flat file. The logic essentially asks the software to check the row position and, based on its parity, decide whether to mirror that data in a new location or leave the destination cell empty.

Furthermore, this technique is highly adaptable. While the primary goal may be to move every

second row, the same logic can be extended to move every third, fourth, or N-th row by using the **MOD function**. However, for the majority of use cases involving paired data points--such as "Product Name" followed by "Sales Figure"--the simple binary distinction between even and odd rows is sufficient to achieve a clean, structured result that is ready for **auditing** or presentation to stakeholders.

Core Formulas for Segmenting Data Rows

Fortunately, executing this data transformation is straightforward when utilizing the correct **Excel formulas**. There are two primary formulas you will need to master depending on whether your target data begins on an even or odd row. These formulas serve as the engine for your data reorganization, automating what would otherwise be a tedious manual task.

The first formula is designed to capture data located on even-numbered rows. It utilizes the **ISEVEN function** wrapped inside an **IF function**. This setup checks the current row index and, if it finds an even number, retrieves the content from the specified cell. If the row is odd, the formula is instructed to return a null string, effectively leaving the cell blank and maintaining the vertical alignment of the dataset.

The second formula performs the inverse operation by focusing on odd-numbered rows. By employing the **ISODD function**, the user can isolate the alternate set of data. This dual-formula approach provides complete control over the **data wrangling** process, allowing for the creation of two distinct columns that represent the original interleaved data. Below are the specific syntax structures for these operations:

Formula 1: Move All Even-Numbered Rows to Column

=IF(ISEVEN(ROW(B2)),B2,"")

Formula 2: Move All Odd-Numbered Rows to Column

=IF(ISODD(ROW(B3)),B3,"")

Practical Application: Moving Even-Numbered Rows

To better understand how these formulas operate in a real-world scenario, let us examine a practical **example**. Consider a situation where a **spreadsheet** contains a list of sales figures for various products. In this dataset, the product names and their corresponding sales numbers are stacked in a single column, which makes performing **statistical analysis** difficult. The objective is to separate these figures into a dedicated "Sales" column for better readability.

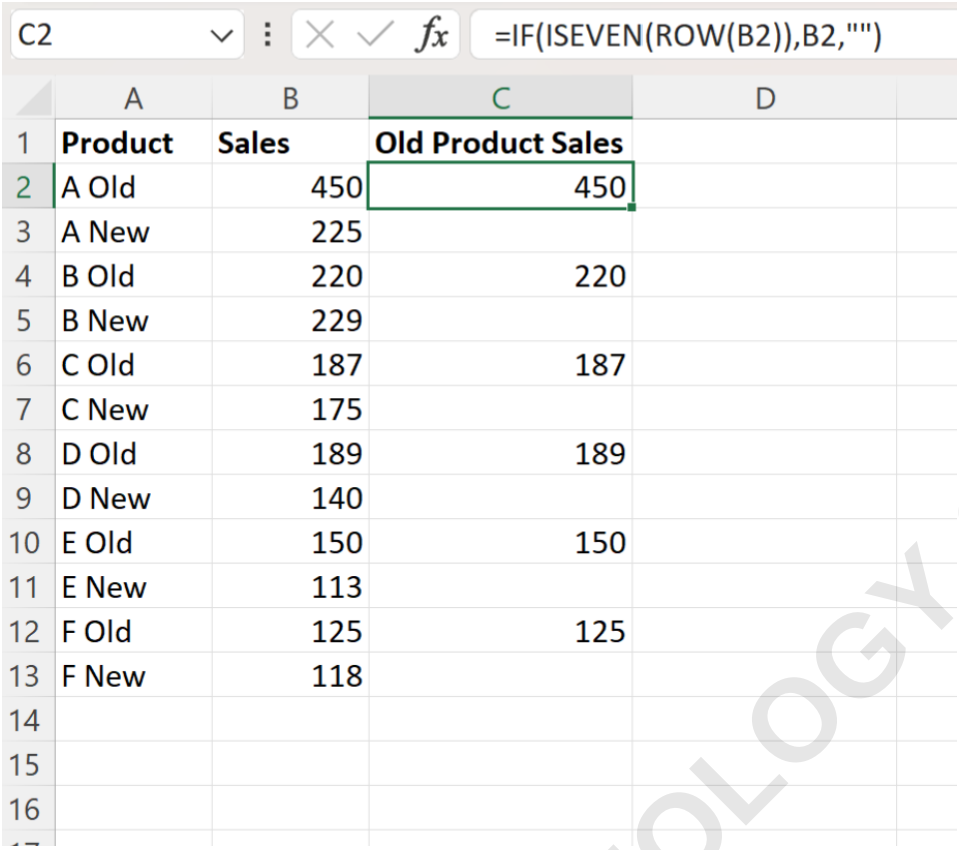
	A	B	C	D	E
1	**Product**	**Sales**			
2	A Old	450			
3	A New	225			
4	B Old	220			
5	B New	229			
6	C Old	187			
7	C New	175			
8	D Old	189			
9	D New	140			
10	E Old	150			
11	E New	113			
12	F Old	125			
13	F New	118			
14					
15					

Suppose we wish to move every other row from the primary data column into a new column (Column C). Specifically, we want to extract the values associated with the "Old" version of products, which happen to reside on even-numbered rows. By positioning our cursor in the first destination cell, we can implement the **ROW function** logic to automate the extraction process across the entire range.

By entering the even-row formula into cell C2, the software evaluates the **metadata** of that specific row. If the condition is met, the value is populated. This creates a filtered view of the data that remains dynamically linked to the original source. This **dynamic reference** is beneficial because any updates made to the original column will automatically reflect in the new column, ensuring your **data analysis** remains accurate and up to date.

=IF(ISEVEN(ROW(B2)),B2,"")

Once the formula is entered, the user can use the **fill handle** to drag the formula down to the remaining cells in the column. This action replicates the logic for every row in the dataset, resulting in a column where only the even-row data is visible, while the odd rows are represented by empty cells, as seen in the following illustration:



	A	B	C	D
1	**Product**	**Sales**	**Old Product Sales**	
2	A Old	450	450	
3	A New	225		
4	B Old	220	220	
5	B New	229		
6	C Old	187	187	
7	C New	175		
8	D Old	189	189	
9	D New	140		
10	E Old	150	150	
11	E New	113		
12	F Old	125	125	
13	F New	118		
14				
15				
16				

Technical Deep Dive: The ISEVEN and ROW Mechanics

The effectiveness of this method lies in the seamless integration of two specific **Excel functions**: **ISEVEN** and **ROW**. The **ROW** function is a reference tool that returns the vertical index of a cell. For example, if you use **ROW(B2)**, the function simply returns the number 2. This numerical output is then passed to the **ISEVEN** function, which performs a **boolean** check to determine if the number is divisible by two without a remainder.

The **IF function** then acts as the decision-maker for the spreadsheet. It takes the "TRUE" or "FALSE" result from the **ISEVEN** check and determines the final output of the cell. If the result is TRUE, it pulls the data from the source cell. If FALSE, it displays a blank. This hierarchical execution of functions is a staple of **advanced Excel modeling** and allows for sophisticated data sorting without the need for complex **VBA scripts** or external plugins.

Understanding this logic also allows users to troubleshoot potential issues. For instance, if your data starts on a different row, you may need to adjust the formula to ensure you are capturing the correct "other" row. This flexibility is what makes **Microsoft Excel** such a versatile tool for **information management**. By manipulating these basic building blocks, you can create customized workflows that cater to the specific structure of any dataset you encounter.

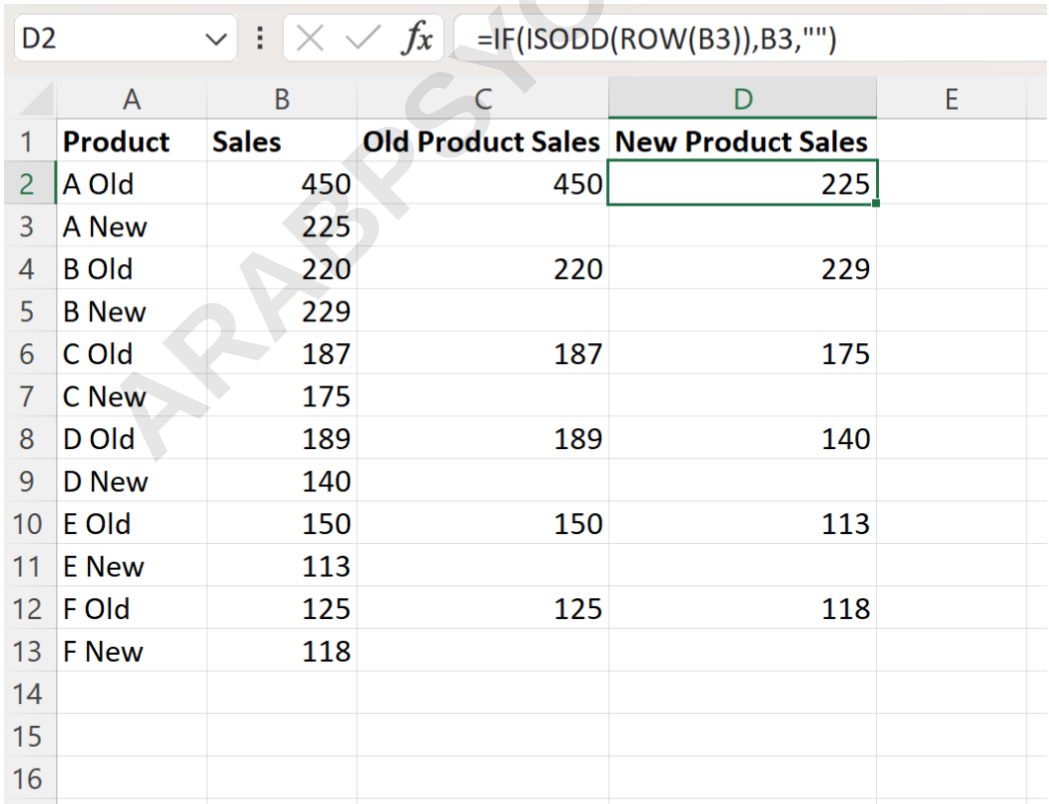
Extracting Odd-Numbered Rows for Comprehensive Analysis

In many instances, isolating only one set of rows is insufficient; you may need to move the other set of alternating rows into a different column as well. For example, if you have separated "Old Product Sales" into Column C, you might want "New Product Sales" in Column D. This requires using the **ISODD function**, which functions identically to its counterpart but targets rows 1, 3, 5, and so on.

To execute this, you would navigate to the next available column and enter the odd-row formula. This ensures that the data points that were previously ignored by the even-row check are now captured. This method effectively "unpacks" the single column of data into a two-column **relational format**, which is much more conducive to generating **charts** and **graphs** for executive summaries.

=IF(ISODD(ROW(B3)),B3,"")

Applying this formula across the range completes the transformation. As seen in the example below, the "New Product Sales" are now isolated in Column D. This creates a clean, structured environment where the relationship between the different data points is visually obvious. Such a layout is the hallmark of a well-maintained **spreadsheet** and is essential for any rigorous **data analysis** project.



	A	B	C	D	E
1	**Product**	**Sales**	**Old Product Sales**	**New Product Sales**	
2	A Old	450	450	225	
3	A New	225			
4	B Old	220	220	229	
5	B New	229			
6	C Old	187	187	175	
7	C New	175			
8	D Old	189	189	140	
9	D New	140			
10	E Old	150	150	113	
11	E New	113			
12	F Old	125	125	118	
13	F New	118			
14					
15					
16					

Refining the Spreadsheet for Professional Presentation

After successfully moving every other row to new columns using the **IF function**, the resulting dataset will contain numerous blank cells. While the data is now organized into columns, these gaps can be distracting and may interfere with certain types of **data processing**. To finalize the spreadsheet, it is often necessary to clean up these blanks to create a contiguous block of information.

One common technique to refine the output is to use the "Go To Special" feature in **Excel** to select and remove blank cells, or to use the **Filter** tool to hide them. Alternatively, if you no longer need the dynamic link to the original data, you can copy the new columns and use **Paste Special > Values** to convert the formulas into static text. This allows you to sort the data or remove the empty rows without affecting the extracted values.

By taking these extra steps, you transform a raw data dump into a professional-grade **report**. This level of attention to detail ensures that the **visualized data** is easy for readers to digest and interpret. Whether you are preparing a financial statement or a sales performance review, using these techniques to move and organize rows will significantly improve the quality of your work and the efficiency of your **workflow**.

Alternative Techniques for Data Transposition

While the formula-based approach using **ISEVEN** and **ISODD** is highly effective, it is not the only way to rearrange data in **Excel**. For users who prefer built-in tools over manual formulas, the **Transpose** feature offers a quick alternative for rotating data from rows to columns. However, standard transposition moves an entire range, which may not always align with the "every other row" requirement without some prior sorting.

Another powerful option for more complex data restructuring is **Power Query**. This tool, built into modern versions of **Excel**, allows for sophisticated **ETL (Extract, Transform, Load)** operations. In Power Query, you can add an **index column**, use the **modulo operator** to identify alternating rows, and then **pivot** the data into the desired format. While this has a steeper learning curve, it is incredibly efficient for handling massive datasets that would slow down a standard worksheet.

Regardless of the specific method chosen, the goal remains the same: to create a clear and actionable **spreadsheet**. Understanding a variety of techniques--from simple **IF statements** to advanced **Power Query** workflows--empowers users to handle any data challenge with confidence and precision.

Summary of Best Practices for Excel Data Management

Mastering the ability to move every other row to a column is just one of many skills that contribute to becoming an **Excel** expert. By integrating logical functions and maintaining a focus on **data structure**, you can significantly enhance the utility of your spreadsheets. Always remember to verify your row indices and test your formulas on a small sample of data before applying them to an entire project to ensure accuracy.

To further improve your proficiency, consider exploring the following areas of **Microsoft Support** and tutorials:

Advanced IF functions and formula nesting techniques.
Using **Conditional Formatting** to highlight alternate rows for better readability.
The **INDEX** and **MATCH** combination for more dynamic data retrieval.
Utilizing **PivotTables** to summarize large volumes of reorganized data.

By following these guidelines and utilizing the formulas discussed, you will be well-equipped to manage and present your data in a structured, professional manner. These skills are invaluable for **data analysis** and ensure that your spreadsheets serve as effective tools for decision-making and communication within your organization.