

How to Select Every Other Column in Excel Easily

Authored by
stats writer

February 19, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Select Every Other Column in Excel Easily*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=131534>

Select Every Other Column in Excel (With Example)

In the contemporary landscape of **data analysis** and business intelligence, Microsoft Excel remains an indispensable tool for professionals who require precision and efficiency. Often, users find themselves navigating through vast datasets where only specific subsets of information are required for **reporting** or **visualization**. One common requirement is the ability to select or isolate every other column within a **spreadsheet**. This task can be performed manually, but for those seeking a more robust and scalable solution, utilizing specialized **functions** is highly recommended.

The introduction of **dynamic arrays** in modern versions of Microsoft 365 has revolutionized how we interact with cell ranges. Specifically, the **CHOOSECOLS** function offers a streamlined approach to extracting specific columns from a **data array** without the need for complex **VBA** scripts or repetitive manual clicking. By defining the specific indices of the columns you wish to retain, you can create a dynamic, linked version of your data that updates automatically as the source **range** changes.

Using the **CHOOSECOLS** function in Excel allows you to select every other column in a specific range with surgical accuracy. This is particularly advantageous when dealing with **financial models**, scientific data, or any **tabular** format where alternating columns represent different categories of information, such as "Projected vs. Actual" figures or "Date vs. Metric" pairs. By mastering this function, you significantly reduce the risk of human error associated with manual selection.

For example, you can use the following formula to select every other column in the range **A1:G11**:

```
=CHOOSECOLS(A1:G11, {1,3,5,7})
```

The primary advantage of this **formula-based** approach is its inherent flexibility. Unlike manual selection, which is static, a formula using the CHOOSECOLS function remains responsive to changes in the underlying **worksheet**. If the values within the source **range** are modified, the output range will immediately reflect those updates, ensuring that your **analysis** remains current without further intervention.

Furthermore, this method facilitates cleaner **data presentation**. Instead of hiding columns--which can often lead to confusion or errors when copying and pasting--you can generate a secondary table that only contains the pertinent **data points**. This keeps your original **dataset** intact while providing a streamlined view for stakeholders or further **statistical calculations**. The following example shows how to use this formula in practice.

Understanding the CHOOSECOLS Function Syntax

To effectively utilize the **CHOOSECOLS** function, one must first understand its **syntax** and **arguments**. The function requires at least two components: the **array** (the source data) and one or more column numbers indicating which specific columns to return. In our context, we provide these numbers as an **array constant** using curly braces, such as {1, 3, 5}. This tells Excel to look at the source **range** and pull only the first, third, and fifth columns into the new location.

This function is part of the **lookup and reference** category of functions and is designed to handle **spilled array** behavior. When you enter the formula into a single cell, **Excel** automatically "spills" the results into the adjacent cells, filling the necessary rows and columns to match the requested data. This eliminates the old-fashioned need for **Ctrl+Shift+Enter** (CSE) formulas, making **spreadsheet** management much more intuitive for the average user.

It is also worth noting that the **column index** numbers you provide are relative to the range itself, not the **worksheet** as a whole. For instance, if your range starts at column C, a column index of 1 refers to column C, not column A. This distinction is crucial for maintaining **data integrity** when working with partial segments of a larger dataset. Understanding this relative positioning allows for more complex **nested formulas** and advanced **data manipulation**.

Example: Select Every Other Column in Excel

Suppose we have the following dataset in **Excel** that contains information about various basketball players, including names, positions, points, rebounds, and other **performance metrics**:

	A	B	C	D	E	F	G
1	Team	Points	Assists	Rebounds	Steals	Blocks	Turnovers
2	Mavs	22	4	5	3	1	5
3	Spurs	19	9	5	0	0	3
4	Rockets	15	3	8	0	0	3
5	Kings	15	8	7	2	2	7
6	Warriors	29	12	9	3	1	5
7	Nets	24	10	12	1	5	1
8	Lakers	40	8	7	2	4	0
9	Thunder	35	3	10	2	2	3
10	Blazers	23	6	5	2	0	0
11	Jazz	33	2	14	1	0	2
12							
13							
14							
15							
16							
17							

In this scenario, the **spreadsheet** contains several columns of data, but for our specific **data analysis** needs, we only want to view every other column to simplify the **visualization**. Suppose we would like to select every other column in this range to create a summarized view for a coaching report. This allows the viewer to focus on high-level **statistics** without being overwhelmed by the granular detail contained in the intermediate columns.

Manual selection in this case would involve holding the **Ctrl** key while clicking each header, which is prone to error if the **dataset** expands to dozens or hundreds of columns. By using a **logical formula**, we ensure that the selection process is consistent and repeatable. This is a hallmark of professional **data management**, where **automation** is prioritized over manual labor to ensure accuracy and save time during the work week.

We can type the following formula into cell **A14** to do so:

=CHOOSECOLS(A1:G11, {1,3,5,7})

The following screenshot shows how to use this formula in practice. Observe how the formula is entered in a single cell, yet it populates a full **array** of data below the original table. This demonstrates the power of **spill ranges** in modern versions of Microsoft Excel. By targeting columns 1, 3, 5, and 7, we effectively bypass the information contained in the even-numbered columns of our source **range**.

	A	B	C	D	E	F	G
1	Team	Points	Assists	Rebounds	Steals	Blocks	Turnovers
2	Mavs	22	4	5	3	1	5
3	Spurs	19	9	5	0	0	3
4	Rockets	15	3	8	0	0	3
5	Kings	15	8	7	2	2	7
6	Warriors	29	12	9	3	1	5
7	Nets	24	10	12	1	5	1
8	Lakers	40	8	7	2	4	0
9	Thunder	35	3	10	2	2	3
10	Blazers	23	6	5	2	0	0
11	Jazz	33	2	14	1	0	2
12							
13							
14	Team	Assists	Steals	Turnovers			
15	Mavs	4	3	5			
16	Spurs	9	0	3			
17	Rockets	3	0	3			
18	Kings	8	2	7			
19	Warriors	12	3	5			
20	Nets	10	1	1			
21	Lakers	8	2	0			
22	Thunder	3	2	3			
23	Blazers	6	2	0			
24	Jazz	2	1	2			
25							

We can see that the **CHOOSECOLS** function returned columns in positions **1, 3, 5** and **7** from the range **A1:G11**. This transformation is instantaneous and creates a live link between the source **data** and the filtered output. If you were to change a player's name in column A, the corresponding cell in the **CHOOSECOLS** output would update immediately, reflecting the change across your entire **workbook**.

Analyzing the Resulting Data Array

Once the **formula** has been executed, the resulting **dataset** is cleaner and more focused. This has the effect of selecting every other column in the range, allowing for a comparative **analysis** of specific variables. In our basketball example, this might mean looking at the "Player Name," "Total Points," "Assists," and "Steals," while skipping over "Position" or "Team Name." Such **data filtering** is essential when preparing data for **pivot tables** or charts.

The clarity provided by this method cannot be overstated. When a **spreadsheet** becomes cluttered with too many columns, the human eye struggles to track patterns. By isolating every other column, you create "white space" for your **analysis**, making it easier to identify outliers or trends in the **basketball statistics**. This is a common technique used in **business intelligence** to create executive summaries that highlight key **performance indicators** (KPIs).

This has the effect of selecting every other column in the range:

	A	B	C	D	E	F	G
1	Team	Points	Assists	Rebounds	Steals	Blocks	Turnovers
2	Mavs	22	4	5	3	1	5
3	Spurs	19	9	5	0	0	3
4	Rockets	15	3	8	0	0	3
5	Kings	15	8	7	2	2	7
6	Warriors	29	12	9	3	1	5
7	Nets	24	10	12	1	5	1
8	Lakers	40	8	7	2	4	0
9	Thunder	35	3	10	2	2	3
10	Blazers	23	6	5	2	0	0
11	Jazz	33	2	14	1	0	2
12							
13							
14	Team	Assists	Steals	Turnovers			
15	Mavs	4	3	5			
16	Spurs	9	0	3			
17	Rockets	3	0	3			
18	Kings	8	2	7			
19	Warriors	12	3	5			
20	Nets	10	1	1			
21	Lakers	8	2	0			
22	Thunder	3	2	3			
23	Blazers	6	2	0			
24	Jazz	2	1	2			
25							
26							

As shown in the image above, the extracted data maintains the **formatting** (if applied to the destination cells) and the structural **integrity** of the original table. Because the first number that we specified in the **CHOOSECOLS** function was **1**, it returned every other column from the range starting with the first column. This is often referred to as selecting the "odd" columns in a 1-based **indexing system**.

If your specific **workflow** requires you to capture the alternate set of columns--the "even" columns--

-the process is just as simple. You simply adjust the **array constant** within the function to target indices 2, 4, 6, and so on. This level of control is what makes **Excel** such a versatile tool for **data scientists** and **accountants** who must manipulate data structures on a daily basis.

Selecting Alternate Column Sets Starting from the Second Column

There are many instances where the most valuable **data points** do not reside in the first column. For example, if the first column contains a unique ID that is not necessary for your current **report**, you may want to start your selection from the second column. This is easily achieved by modifying the **arguments** within our **dynamic array formula**. By shifting our index numbers, we tell Excel to skip the first column and focus on the secondary set of alternating data.

If you would instead like to return every other column starting with the second column, you can use the following formula instead:

```
=CHOOSECOLS(A1:G11, {2,4,6})
```

By using this variation, you extract columns 2, 4, and 6. In a standard **database** export, these might represent "Category," "Price," and "Quantity," while skipping "ID," "Description," and "Supplier." This targeted extraction is vital for **data cleaning** and preparation, ensuring that only the relevant **variables** are passed into your **computational models** or **graphs**.

The following screenshot shows how to use this formula in practice. Notice how the output table now begins with what was originally the second column of our source **range**. This flexibility demonstrates that the **CHOOSECOLS** function is not just for selecting "every other" column in a fixed pattern, but can be used to select any arbitrary set of columns in any order you require for your **documentation**.

This **nested formula** approach is the pinnacle of **automated data selection**. It allows you to select every other column across a massive **range** with a single line of code. If your **worksheet** expands horizontally, the **SEQUENCE** function can be adjusted to accommodate the new width, making your **Excel model** truly scalable and future-proof. This is particularly useful for **time-series analysis** where new columns are added for each passing month or quarter.

Furthermore, for users who prefer not to use formulas, **Power Query** offers a graphical user interface to perform similar tasks. By importing your table into the [Power Query Editor](#), you can add an "Index Column" and then filter the index to keep only rows (which represent columns after a transpose) that are **even** or **odd**. However, for most daily tasks, the **CHOOSECOLS** function remains the fastest and most direct method available within the **Excel grid**.

Manual Methods and Keyboard Shortcuts

Although **formulas** provide the most dynamic solution, there are times when a quick, manual selection is necessary for a one-off task. As mentioned in the original **tutorial**, you can use **keyboard shortcuts** to navigate and select columns. To select non-adjacent columns, you must utilize the **Ctrl** key. First, click the header of the first column you wish to select, then hold down **Ctrl** while clicking the headers of every subsequent column you need. This highlights the columns for **copying, deleting, or formatting**.

For keyboard enthusiasts, you can use **Ctrl + Space** to select an entire column. After selecting the first one, you can move to the next desired column using the arrow keys and then use **Shift + F8** to enter "Add to Selection" mode. This allows you to pick up additional columns without losing your previous selection. While this is faster than using a mouse, it still lacks the **reusability** and **dynamic nature** of the **CHOOSECOLS** function.

Another manual approach involves using **Excel Tables**. When your data is formatted as a [Table \(Ctrl+T\)](#), you can use the "Banded Columns" option under the Table Design tab. While this doesn't "select" the columns in terms of isolation, it applies **shading** to every other column, which provides the necessary **visual distinction** to help you manually identify and work with alternating sets of data more effectively.

Best Practices for Data Integrity and Formatting

When extracting data using **CHOOSECOLS**, it is important to consider the **formatting** of the destination cells. **Excel formulas** typically return the raw values or **data types**, but they do not always carry over the specific **cell styles**, such as borders or background colors, from the source. To maintain a professional look, you may need to apply **Conditional Formatting** to your output range or use the **Format Painter** to match the aesthetic of your original **spreadsheet**.

Additionally, always verify the **source range** reference. If you use a **static reference** like A1:G11, your formula will not account for new rows added at the bottom. To solve this, consider converting your source data into an **Excel Table** and using **structured references** (e.g., Table1). This ensures that as you add more basketball players to your list, the **CHOOSECOLS** function will automatically include the new **data points** in its output, maintaining the **integrity** of your report.

Finally, be mindful of the **#SPILL! error**. This occurs when there is existing data in the cells where the **CHOOSECOLS** function is trying to display its results. Always ensure that the area below and to the right of your formula is clear of other **content**. This allows the **dynamic array** to expand freely, providing you with a complete and uninterrupted view of your selected **columns**.

Note: You can find the complete documentation for the **CHOOSECOLS** function on the [official Microsoft Support website](#), which provides additional details on **error handling** and usage limits.

Summary of Common Excel Operations

The ability to select every other column is just one of many **techniques** used to master **data management** in Excel. Whether you are performing a **VLOOKUP**, managing **Pivot Tables**, or utilizing **Power Query**, the goal is always to transform raw information into actionable **insights**. Mastering these **functions** allows you to handle increasingly complex **datasets** with ease and confidence.

The following tutorials explain how to perform other common operations in **Excel**, helping you further expand your **technical proficiency** and streamline your daily **workflows**:

How to use the **CHOOSEROWS** function for vertical data extraction.

Methods for **filtering data** based on cell color or font style.

Advanced **conditional formatting** for alternating rows and columns.

Using **INDEX** and **MATCH** as a powerful alternative to **VLOOKUP**.

Automating repetitive tasks with **Excel Macros** and **VBA**.

By integrating these **skills**, you become a more effective **data analyst**, capable of providing high-quality **business intelligence** in any professional setting. Excel's **dynamic array functions**, like **CHOOSECOLS**, are the key to modern, efficient **spreadsheet** design.