

# How to Easily Select Every Nth Row in Excel

Authored by  
**stats writer**

November 30, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Easily Select Every Nth Row in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102283>

When efficiently managing large datasets in Excel, a common requirement is the ability to selectively extract data, such as selecting every Nth row. This capability is vital for tasks like sampling, reporting, or reducing the granularity of massive data tables. Achieving this precise selection requires leveraging powerful dynamic array formulas, specifically combining the functionality of the OFFSET function with the ROW function.

The core mechanism involves using the OFFSET function, which dynamically adjusts a reference based on specified row and column offsets from a starting point. By integrating the ROW function, we generate a calculated offset that increases geometrically, ensuring that only the desired Nth rows are targeted. The foundational formula used to accomplish this is structured as `OFFSET(A1, (ROW() - 1) * n, 0)`. This formula reliably selects every Nth row, beginning with the initial row specified in the anchor reference (A1).

## Understanding the Need for Selective Data Extraction

When dealing with extensive spreadsheets, manually identifying and extracting rows based on a specific numerical interval (N) is not only time-consuming but highly prone to error. Imagine needing to pull only the first entry of every fifty records for a statistical survey; this task necessitates an automated, precise solution. Selective extraction allows analysts to create representative samples efficiently, aiding in performance testing, validating data subsets, or simplifying reports that only require periodic data points rather than continuous streams.

The fundamental challenge in Excel lies in creating a formula that understands its own position relative to the starting cell and calculates the required jump dynamically. Unlike simple absolute referencing, selecting every Nth row requires a mathematical progression that resets and recalculates for each cell where the formula is copied. We must program the formula to recognize that in the first cell it should return row 1, in the second cell it should return row (1+N), in the third cell it should return row (1+2N), and so forth.

To address this sophisticated requirement, the combination of functions must generate a series of row offsets that are multiples of the chosen interval 'N'. The formula serves as an indexing mechanism, iterating through the source data at fixed steps. This ensures data integrity and consistency, drastically reducing the labor involved in preparing large data segments for analysis or further processing outside of the main spreadsheet environment.

## The Core Formula: OFFSET and ROW Functions

The power of this technique resides in the synergy between the OFFSET function and the ROW function. The general structure provides the foundation for selecting every Nth row in your data table. Specifically, you will use the following structure, where **n** represents the desired interval

(e.g., 3 for every third row):

You can use the following basic formula to select every nth row in Excel:

**=OFFSET(\$A\$1,(ROW()-1)\*n,0)**

This formula selects every **nth** row.

Understanding the components of this formula is critical for successful implementation and modification. The formula utilizes absolute referencing on the starting cell (\$A\$1) to ensure the anchor point remains constant when the formula is copied down. The dynamic element,  $(\text{ROW}() - 1) * n$ , is where the magic happens, calculating the exact number of rows Excel must jump or offset from the fixed starting point to reach the target Nth row.

By simply changing the integer value assigned to n within the formula, you instantly adjust the sampling interval. For instance, if you wish to extract data only from every **third** row, you substitute n with 3. If you need every fifth row, you substitute n with 5. This flexibility makes the method highly adaptable across various reporting requirements and dataset sizes.

## Deconstructing the OFFSET Function Syntax

To master this extraction technique, a thorough understanding of the OFFSET function's arguments is necessary. The OFFSET syntax is defined as `OFFSET(reference, rows, cols, , )`. In our application, we only utilize the first three arguments: reference, rows, and cols.

**Reference (\$A\$1):** This is the starting point, locked using absolute referencing. It defines the cell from which the offset calculation begins.

**Rows ((ROW() - 1) \* n):** This is the calculated number of rows to move down (positive value) or up (negative value) from the reference cell.

**Cols (0):** This specifies the number of columns to move. Since we want to stay in the same column as our source data (Column A), we set this value to **zero**.

The most crucial part of the dynamic formula is the rows argument:  $(\text{ROW}() - 1) * n$ . When this formula is entered into cell C1 (the first output cell), the ROW function returns 1. The calculation becomes  $(1 - 1) * n$ , resulting in 0. The offset is 0 rows, so the formula returns the value in \$A\$1. When copied down to cell C2, ROW() returns 2. The calculation becomes  $(2 - 1) * n$ , resulting in  $1 * n$ . If  $n=3$ , the offset is 3 rows, selecting \$A\$4. This iterative process ensures that the selected row index consistently increments by N.

## Practical Example 1: Selecting Every Third Row

To illustrate the application of this powerful formula, let us consider a practical scenario where we need to retrieve only every **third** data point from a continuous list. This is often necessary when analyzing time-series data or performance metrics where only periodic checks are relevant.

For example, you can use the following formula to select every **3rd** row:

**=OFFSET(\$A\$1,(ROW()-1)\*3,0)**

In this specific instance, setting *n* to 3 dictates that the row offset will be 0, 3, 6, 9, 12, and so on, relative to the starting cell **\$A\$1**. The formula in the output column (Column C) will pull the values from rows 1, 4, 7, 10, 13, etc., of the source column (Column A). This mechanism effectively filters the source dataset based on the defined periodicity.

The key advantage here is scalability. Once the interval (*N*=3) is defined in the first cell, you do not need to manually adjust any other parameters. The relative nature of the ROW function ensures that when you drag the formula down the column, the offset calculation self-adjusts, maintaining the third-row selection pattern seamlessly across thousands of rows of data.

## Step-by-Step Implementation Guide (N=3)

The following examples show how to use this formula in practice.

Suppose we begin with a simple list of numerical values located in Column A of our Excel sheet. Our goal is to extract every third row into a separate column, Column C. This structured approach ensures accuracy and reproducibility of the results.

Suppose we have the following list of values in Excel:

|    | A  | B | C | D | E | F |
|----|----|---|---|---|---|---|
| 1  | 4  |   |   |   |   |   |
| 2  | 8  |   |   |   |   |   |
| 3  | 12 |   |   |   |   |   |
| 4  | 14 |   |   |   |   |   |
| 5  | 19 |   |   |   |   |   |
| 6  | 22 |   |   |   |   |   |
| 7  | 24 |   |   |   |   |   |
| 8  | 28 |   |   |   |   |   |
| 9  | 29 |   |   |   |   |   |
| 10 | 31 |   |   |   |   |   |
| 11 | 34 |   |   |   |   |   |
| 12 | 35 |   |   |   |   |   |
| 13 | 49 |   |   |   |   |   |
| 14 | 13 |   |   |   |   |   |
| 15 | 6  |   |   |   |   |   |
| 16 | 7  |   |   |   |   |   |
| 17 | 12 |   |   |   |   |   |
| 18 | 10 |   |   |   |   |   |
| 19 | 11 |   |   |   |   |   |
| 20 | 6  |   |   |   |   |   |
| 21 |    |   |   |   |   |   |
| 22 |    |   |   |   |   |   |

First, identify the target cell where the extracted data should begin. In this example, we will start the output in cell **C1**. We input the formula designed for selecting every third row, ensuring that the reference is locked to the absolute cell **\$A\$1**. This establishes the necessary anchoring for the subsequent dynamic calculations.

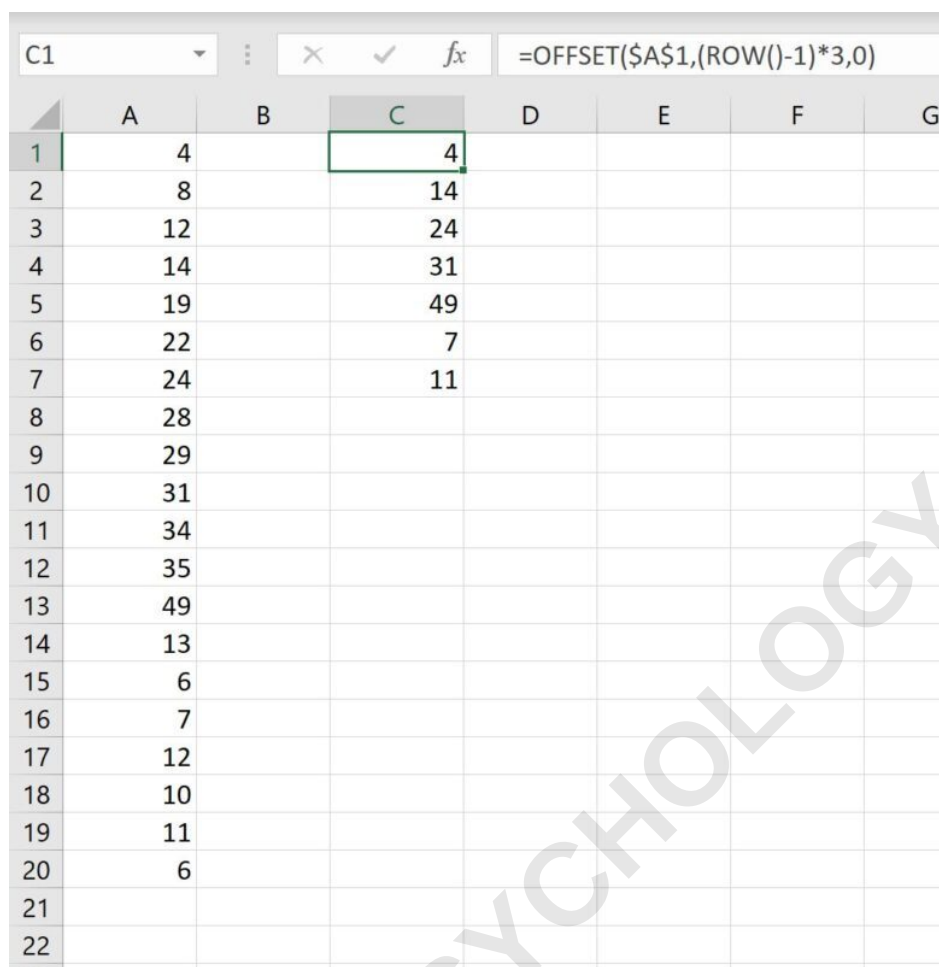
We can use the following formula to select every **third** row from the list:

**=OFFSET(\$A\$1,(ROW()-1)\*3,0)**

### Analyzing the Results of the Third Row Selection

Once the formula is entered into cell **C1**, the next critical step is propagation. We must copy and paste this formula down the remainder of Column C, ensuring that we cover at least as many rows as are present in the original data set in Column A. This action triggers the relative calculation of the ROW function in each subsequent cell, generating the geometrically increasing offsets required by the OFFSET function.

We will type this formula into cell **C1** and then copy and paste it to the remaining cells in column C:



The screenshot shows an Excel spreadsheet with columns A through G and rows 1 through 22. The formula bar at the top displays the formula `=OFFSET($A$1,(ROW()-1)*3,0)` in cell C1. The spreadsheet data is as follows:

|    | A  | B | C  | D | E | F | G |
|----|----|---|----|---|---|---|---|
| 1  | 4  |   | 4  |   |   |   |   |
| 2  | 8  |   | 14 |   |   |   |   |
| 3  | 12 |   | 24 |   |   |   |   |
| 4  | 14 |   | 31 |   |   |   |   |
| 5  | 19 |   | 49 |   |   |   |   |
| 6  | 22 |   | 7  |   |   |   |   |
| 7  | 24 |   | 11 |   |   |   |   |
| 8  | 28 |   |    |   |   |   |   |
| 9  | 29 |   |    |   |   |   |   |
| 10 | 31 |   |    |   |   |   |   |
| 11 | 34 |   |    |   |   |   |   |
| 12 | 35 |   |    |   |   |   |   |
| 13 | 49 |   |    |   |   |   |   |
| 14 | 13 |   |    |   |   |   |   |
| 15 | 6  |   |    |   |   |   |   |
| 16 | 7  |   |    |   |   |   |   |
| 17 | 12 |   |    |   |   |   |   |
| 18 | 10 |   |    |   |   |   |   |
| 19 | 11 |   |    |   |   |   |   |
| 20 | 6  |   |    |   |   |   |   |
| 21 |    |   |    |   |   |   |   |
| 22 |    |   |    |   |   |   |   |

Upon reviewing the results in Column C, a clear pattern emerges. The output column only displays values that correspond to the 1st, 4th, 7th, 10th, and 13th rows of the source data. All rows that were not a multiple of three from the starting point have been intentionally skipped by the calculated offset.

Notice that every third row has been selected from the list of original values:

|    | A  | B | C  | D | E | F | G |
|----|----|---|----|---|---|---|---|
| 1  | 4  |   | 4  |   |   |   |   |
| 2  | 8  |   | 14 |   |   |   |   |
| 3  | 12 |   | 24 |   |   |   |   |
| 4  | 14 |   | 31 |   |   |   |   |
| 5  | 19 |   | 49 |   |   |   |   |
| 6  | 22 |   | 7  |   |   |   |   |
| 7  | 24 |   | 11 |   |   |   |   |
| 8  | 28 |   |    |   |   |   |   |
| 9  | 29 |   |    |   |   |   |   |
| 10 | 31 |   |    |   |   |   |   |
| 11 | 34 |   |    |   |   |   |   |
| 12 | 35 |   |    |   |   |   |   |
| 13 | 49 |   |    |   |   |   |   |
| 14 | 13 |   |    |   |   |   |   |
| 15 | 6  |   |    |   |   |   |   |
| 16 | 7  |   |    |   |   |   |   |
| 17 | 12 |   |    |   |   |   |   |
| 18 | 10 |   |    |   |   |   |   |
| 19 | 11 |   |    |   |   |   |   |
| 20 | 6  |   |    |   |   |   |   |
| 21 |    |   |    |   |   |   |   |
| 22 |    |   |    |   |   |   |   |

### Adjusting N for Different Intervals (Selecting Every Fifth Row)

The real utility of this method is demonstrated by its adaptability. If the analytical requirement changes, and we now need to sample every fifth row instead of every third, the only necessary modification is updating the value of *n* in the core formula. This ability to easily adjust the sampling interval without complex formula rewriting streamlines data processing significantly.

If we change the value of *n* in the formula, we can select a different *n*th value.

To select every fifth row, we substitute *n*=5 into the multiplier section of the formula. This ensures that the offset calculation now jumps in increments of five, targeting rows 1, 6, 11, 16, and so on. This immediate change allows for rapid re-sampling of the same source data, adhering to the new interval mandate.

**=OFFSET(\$A\$1,(ROW()-1)\*5,0)**

We repeat the implementation steps: enter the modified formula (with  $n=5$ ) into cell **C1**, and then copy this formula downwards throughout the output column. Observe how the change in the multiplier drastically alters the output, demonstrating the sensitivity and effectiveness of the  $(\text{ROW}()-1)*n$  component.

We will type this formula into cell **C1** and then copy and paste it to the remaining cells in column C:

|    | A  | B | C  | D | E | F | G |
|----|----|---|----|---|---|---|---|
| 1  | 4  |   | 4  |   |   |   |   |
| 2  | 8  |   | 22 |   |   |   |   |
| 3  | 12 |   | 34 |   |   |   |   |
| 4  | 14 |   | 7  |   |   |   |   |
| 5  | 19 |   |    |   |   |   |   |
| 6  | 22 |   |    |   |   |   |   |
| 7  | 24 |   |    |   |   |   |   |
| 8  | 28 |   |    |   |   |   |   |
| 9  | 29 |   |    |   |   |   |   |
| 10 | 31 |   |    |   |   |   |   |
| 11 | 34 |   |    |   |   |   |   |
| 12 | 35 |   |    |   |   |   |   |
| 13 | 49 |   |    |   |   |   |   |
| 14 | 13 |   |    |   |   |   |   |
| 15 | 6  |   |    |   |   |   |   |
| 16 | 7  |   |    |   |   |   |   |
| 17 | 12 |   |    |   |   |   |   |
| 18 | 10 |   |    |   |   |   |   |
| 19 | 11 |   |    |   |   |   |   |
| 20 | 6  |   |    |   |   |   |   |
| 21 |    |   |    |   |   |   |   |
| 22 |    |   |    |   |   |   |   |
| 23 |    |   |    |   |   |   |   |

## Reviewing the Results of the Fifth Row Selection

By comparing the results of the  $N=3$  selection with the  $N=5$  selection, we clearly see the impact of adjusting the interval. The  $N=5$  output will contain significantly fewer data points extracted from the same source list, as the sampling is less frequent. This technique is especially useful when creating summaries or dashboards that rely only on key milestones within a long sequence of data.

Notice that every fifth row has been selected from the list of original values:

|    | A  | B | C  | D | E | F | G |
|----|----|---|----|---|---|---|---|
| 1  | 4  |   | 4  |   |   |   |   |
| 2  | 8  |   | 22 |   |   |   |   |
| 3  | 12 |   | 34 |   |   |   |   |
| 4  | 14 |   | 7  |   |   |   |   |
| 5  | 19 |   |    |   |   |   |   |
| 6  | 22 |   |    |   |   |   |   |
| 7  | 24 |   |    |   |   |   |   |
| 8  | 28 |   |    |   |   |   |   |
| 9  | 29 |   |    |   |   |   |   |
| 10 | 31 |   |    |   |   |   |   |
| 11 | 34 |   |    |   |   |   |   |
| 12 | 35 |   |    |   |   |   |   |
| 13 | 49 |   |    |   |   |   |   |
| 14 | 13 |   |    |   |   |   |   |
| 15 | 6  |   |    |   |   |   |   |
| 16 | 7  |   |    |   |   |   |   |
| 17 | 12 |   |    |   |   |   |   |
| 18 | 10 |   |    |   |   |   |   |
| 19 | 11 |   |    |   |   |   |   |
| 20 | 6  |   |    |   |   |   |   |
| 21 |    |   |    |   |   |   |   |
| 22 |    |   |    |   |   |   |   |

It is important to note that while the OFFSET function is powerful for this specific task, it is considered a **volatile function**. Volatile functions recalculate every time any cell in the workbook changes, which can slow down performance considerably in extremely large or complex spreadsheets. For modern Excel versions, alternatives like using the INDEX function combined with SEQUENCE and MOD functions might offer better performance for true dynamic array spill behavior, though OFFSET remains the classic approach for compatibility and simplicity in basic row selection tasks.

Ultimately, mastering the dynamic referencing inherent in the OFFSET and ROW combination provides an indispensable tool for data manipulation, allowing analysts to extract precise, systematic samples from voluminous spreadsheets with high confidence and minimal manual intervention.