

# How to Autofill Values in Excel from Another Sheet: A Step-by-Step Guide

Authored by  
**stats writer**

November 28, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Autofill Values in Excel from Another Sheet: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=101145>

The process of managing large datasets often requires integrating information stored across multiple sections or files. In Microsoft Excel, this frequently involves consolidating data from separate worksheets within the same workbook. Utilizing the autofill feature combined with proper cell referencing provides an exceptionally efficient method for linking and dynamically updating data across sheets.

This technique is superior to simple copy-and-paste operations because it establishes a live link between the source data and the destination location. If the original values on the source sheet are modified, the linked values on the destination sheet update automatically, ensuring data integrity and reducing manual intervention. The goal of this guide is to demonstrate how to leverage the autofill feature--specifically the powerful functionality of the Fill Handle--to propagate references from one initial formula across an entire column, thereby pulling data seamlessly from a separate worksheet.

The following detailed, step-by-step example illustrates the precise methodology required to swiftly and accurately autofill values derived from a source worksheet into a destination worksheet in Excel.

## Understanding Cross-Sheet Data Integration in Excel

Cross-sheet data integration is foundational for building complex financial models, dashboards, and detailed reports in Excel. Before applying the autofill feature, it is critical to understand the concept of external cell referencing. When you reference a cell located on a different sheet within the same workbook, the formula structure must include the name of the source sheet, followed by an exclamation mark (the sheet separator), and then the standard cell address.

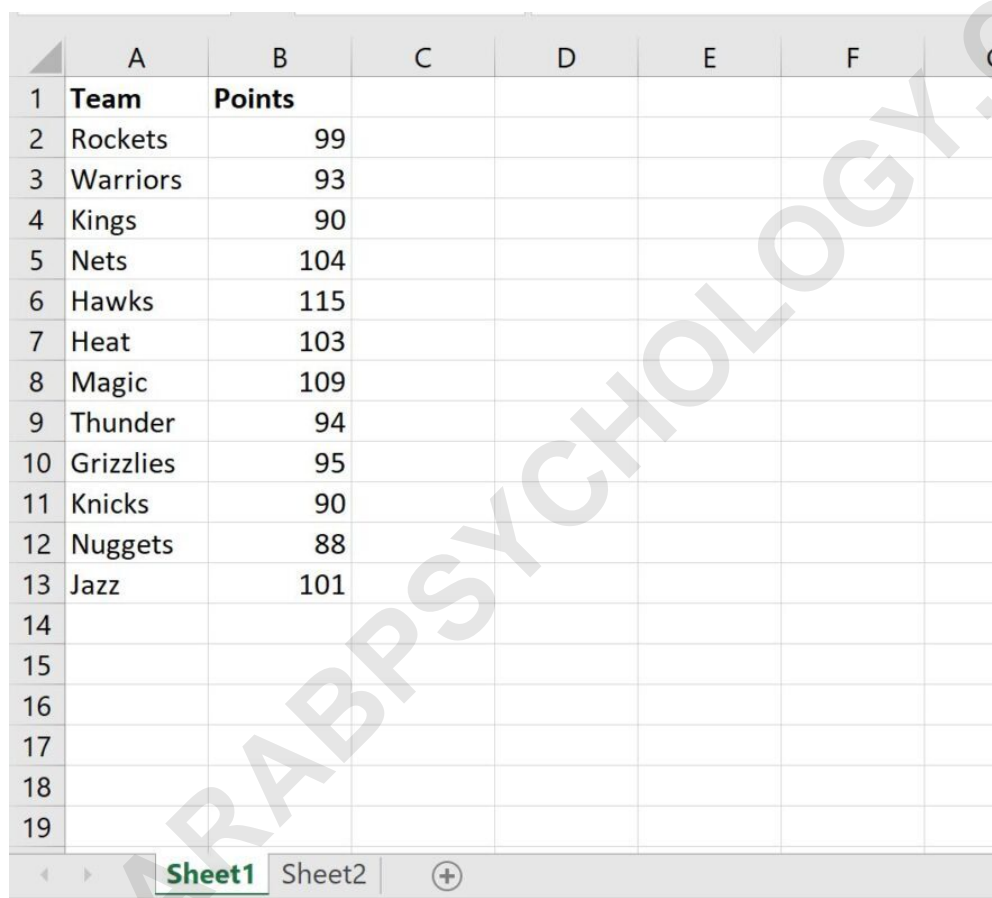
For instance, to retrieve the value from cell **B2** located on a sheet named **Sheet1**, the appropriate formula syntax is `=Sheet1!B2`. This syntax explicitly directs Excel to the exact location of the source data. When using the Fill Handle (the small square in the bottom right corner of the selected cell), Excel's relative referencing capability automatically adjusts the row number (e.g., from B2 to B3, B4, and so on) as the formula is dragged down, ensuring that each cell on the destination sheet correctly links to the corresponding row on the source sheet. This dynamic linking capability is the core strength of this method.

This approach maintains a level of dynamism and flexibility that static data transfer methods lack. If the source data is volatile or subject to frequent revisions, establishing these linked cell references ensures that all subsequent reports and analyses using the destination data are based on the most current information available in the source worksheet.

## Step 1: Preparing the Source Worksheet (Sheet1)

Our initial step involves preparing the source data which we intend to transfer. We will use a sheet named **Sheet1** for this purpose. This sheet contains foundational data, structured typically with headers in Row 1 and subsequent data entries below. For this demonstration, we will focus on extracting a specific column of numerical data.

Let's input the following sample data into **Sheet1** in Excel. Column A contains identifiers (Name) and Column B contains the numerical data we wish to transfer (Points):



	A	B	C	D	E	F	G
1	<b>Team</b>	<b>Points</b>					
2	Rockets	99					
3	Warriors	93					
4	Kings	90					
5	Nets	104					
6	Hawks	115					
7	Heat	103					
8	Magic	109					
9	Thunder	94					
10	Grizzlies	95					
11	Knicks	90					
12	Nuggets	88					
13	Jazz	101					
14							
15							
16							
17							
18							
19							

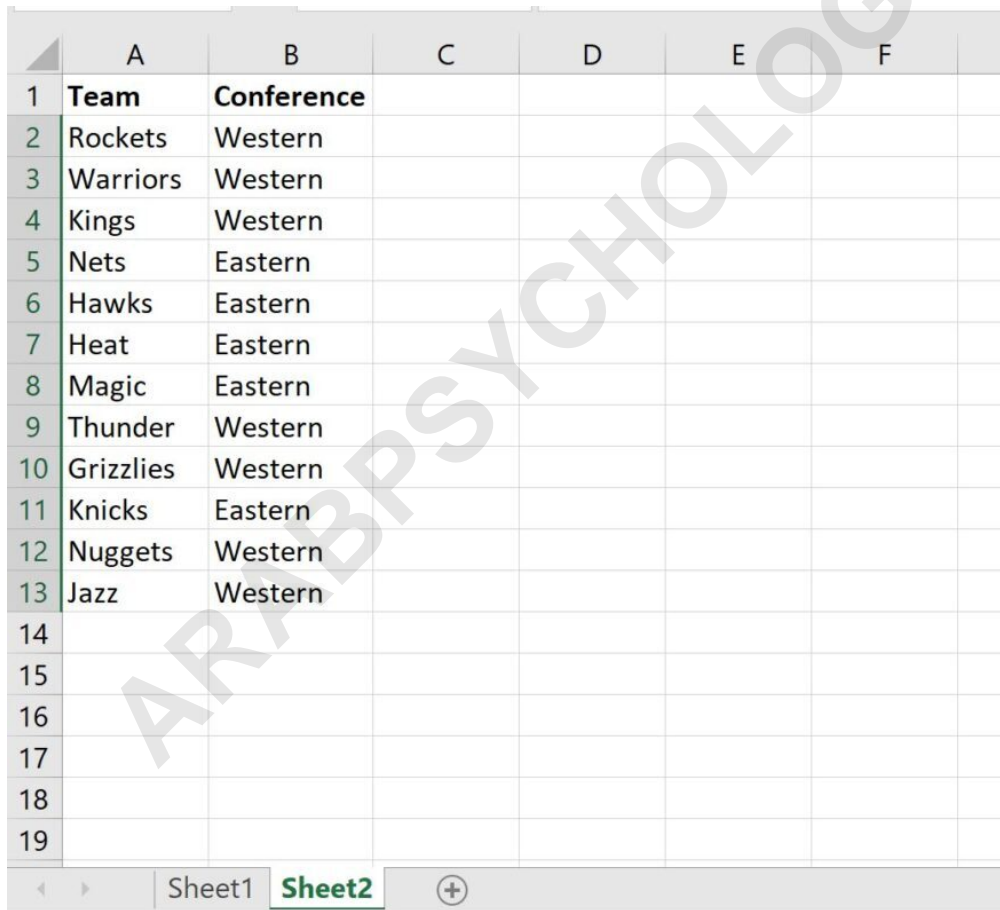
In this setup, the critical data points we need to link start from cell **B2** (the point value for John) and extend down the column. Ensuring the source data is clean, correctly formatted, and consistently structured is paramount before proceeding to the linking stage. Misaligned data in the source sheet will result in incorrect references when using the Fill Handle.

It is good practice to confirm that the row alignment between the two sheets is logical. While Excel's relative referencing handles the vertical movement automatically, the effectiveness of the

autofill feature relies on the corresponding rows having relevant data on both sheets. If the rows are intentionally misaligned, absolute references (using the dollar sign, e.g., \$B\$2) or more advanced lookup functions would be necessary, but for a direct, row-by-row transfer, relative referencing is ideal.

## Step 2: Establishing the Destination Worksheet and Initial Formula (Sheet2)

Next, we pivot our attention to the destination sheet, titled **Sheet2**. This sheet may contain related information or require the data from **Sheet1** to complete its dataset. Suppose **Sheet2** already contains some structural data, perhaps identifiers that match those in **Sheet1**, but lacks the crucial 'Points' data:



	A	B	C	D	E	F
1	<b>Team</b>	<b>Conference</b>				
2	Rockets	Western				
3	Warriors	Western				
4	Kings	Western				
5	Nets	Eastern				
6	Hawks	Eastern				
7	Heat	Eastern				
8	Magic	Eastern				
9	Thunder	Western				
10	Grizzlies	Western				
11	Knicks	Eastern				
12	Nuggets	Western				
13	Jazz	Western				
14						
15						
16						
17						
18						
19						

Our objective is to populate the empty column (Column C, labeled "Points") in **Sheet2** with the corresponding 'Points' values from **Sheet1**. The alignment is straightforward: we want the data for the first entry in **Sheet2** (Row 2) to come from the first entry in **Sheet1** (Row 2), and so on. This

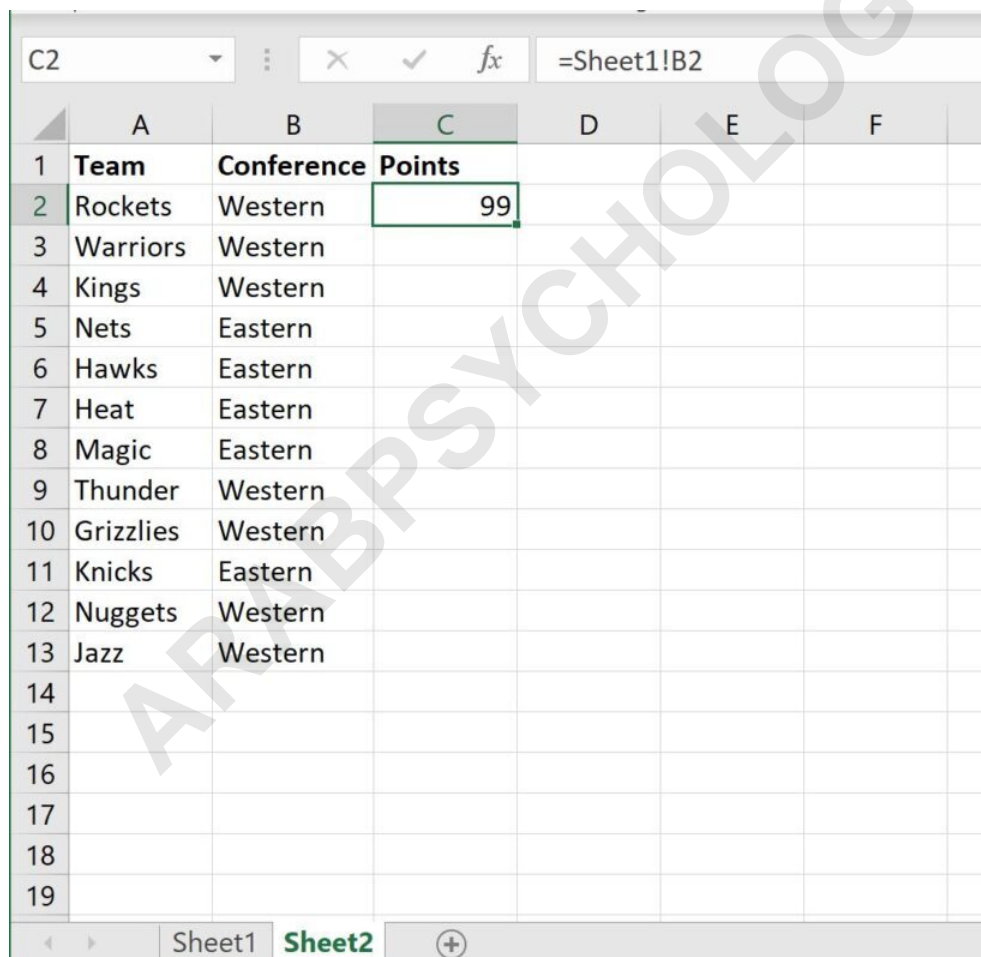
row-by-row correlation is what makes the autofill feature so effective here.

To initiate the transfer, we must first manually establish the cell reference in the very first cell of the destination column. In this example, that cell is **C2** in **Sheet2**. We type the formula that links directly to the data we want to pull, which is cell **B2** on **Sheet1**.

Specifically, we enter the following formula into cell **C2** of **Sheet2**:

**=Sheet1!B2**

This action creates the foundational link. Upon hitting Enter, cell **C2** in **Sheet2** will automatically display the value 95, which is the exact content of cell **B2** in **Sheet1**. This visual confirmation is crucial for verifying that the cell reference syntax is correct and operational:



	A	B	C	D	E	F
1	<b>Team</b>	<b>Conference</b>	<b>Points</b>			
2	Rockets	Western	99			
3	Warriors	Western				
4	Kings	Western				
5	Nets	Eastern				
6	Hawks	Eastern				
7	Heat	Eastern				
8	Magic	Eastern				
9	Thunder	Western				
10	Grizzlies	Western				
11	Knicks	Eastern				
12	Nuggets	Western				
13	Jazz	Western				
14						
15						
16						
17						
18						
19						

## Implementing the Autofill Feature Using the Fill Handle

Once the initial formula is established in cell **C2**, the remaining 99% of the task is handled by Excel's automated autofill feature, accessed via the Fill Handle. The Fill Handle is a small, often overlooked, square visible at the bottom right-hand corner of the active cell selection.

To replicate the formula down column C and pull the remaining values from **Sheet1**, follow these steps:

**Select the Initial Cell:** Ensure cell **C2** (which contains the formula `=Sheet1!B2`) is selected.

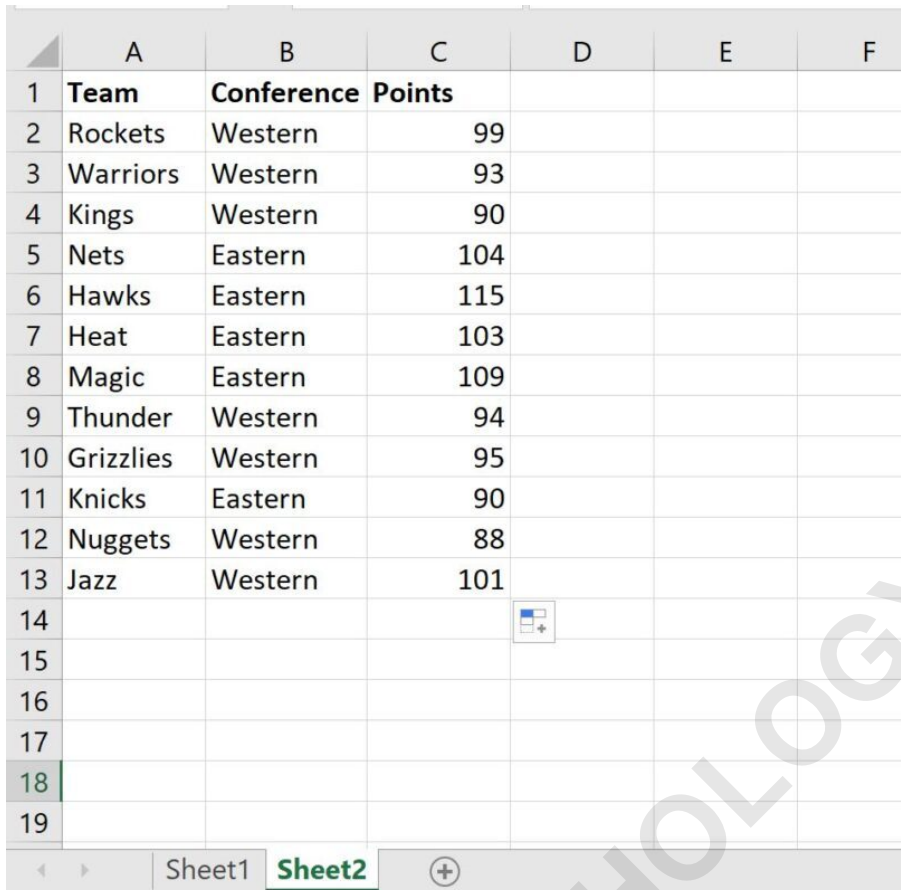
**Locate the Fill Handle:** Hover your mouse cursor over the small green square located at the bottom-right corner of cell **C2**. The cursor will change from a large white cross to a thin black cross (the Fill Handle symbol).

**Execute the Autofill:** You have two primary methods for execution:

**Drag Method:** Click and hold the black cross, then drag it downwards to cover all the rows where you need the data replicated.

**Double-Click Method (Recommended for contiguous data):** If the column immediately to the left (Column B) contains continuous data without any blank rows, simply double-click the black cross. Excel will automatically propagate the formula down the column until it encounters the first blank row in the adjacent column, saving significant time, especially for thousands of rows.

Once you execute the double-click or drag action, Excel employs relative referencing. The formula in cell C3 will become `=Sheet1!B3`, the formula in C4 will be `=Sheet1!B4`, and so on. This ensures a perfect row-by-row data transfer, populating the entire target column dynamically:



	A	B	C	D	E	F
1	<b>Team</b>	<b>Conference</b>	<b>Points</b>			
2	Rockets	Western	99			
3	Warriors	Western	93			
4	Kings	Western	90			
5	Nets	Eastern	104			
6	Hawks	Eastern	115			
7	Heat	Eastern	103			
8	Magic	Eastern	109			
9	Thunder	Western	94			
10	Grizzlies	Western	95			
11	Knicks	Eastern	90			
12	Nuggets	Western	88			
13	Jazz	Western	101			
14						
15						
16						
17						
18						
19						

As demonstrated in the resulting image, all the values from the Points column in **Sheet1** have been successfully and instantaneously autofilled into the corresponding column in **Sheet2**. This operation confirms the power and simplicity of combining cross-sheet referencing with the autofill feature for quick data consolidation.

### Benefits of Direct Autofill vs. Copy/Paste

While manually copying and pasting data might seem faster initially for small datasets, utilizing the autofill feature with linked cell references offers several key advantages that are critical for professional spreadsheet management.

Firstly, the data remains **dynamic**. If the source value (e.g., the score of 95 in Sheet1!B2) is changed to 99, the corresponding cell in Sheet2 (C2) will instantly reflect 99 without any manual intervention. Conversely, if data is simply copied and pasted, it becomes static and disconnected from the source, leading to potential discrepancies if the source sheet is updated later. Maintaining dynamic links is essential for modeling and reporting where underlying data is frequently revised.

Secondly, the process enhances **efficiency and scalability**. For spreadsheets containing hundreds or thousands of rows, the double-click autofill method is significantly faster than dragging the selection or repeatedly copying sections. It eliminates the risk of human error associated with selecting incorrect ranges or stopping too early or too late in the destination column. This efficiency becomes non-negotiable in environments dealing with big data volumes.

Finally, using linked references promotes **data transparency and auditability**. By clicking on a cell in Sheet2, the formula bar clearly shows `=Sheet1!B2`, immediately indicating where the data originated. This level of transparency is invaluable for auditing complex spreadsheets and understanding the flow of data within a workbook, which is often a requirement in compliance and financial reporting.

## Troubleshooting Common Autofill Issues

While the autofill feature is reliable, certain configuration issues can prevent it from working as expected. Addressing these issues ensures seamless operation:

**Issue: Fill Handle is Missing or Disabled.** If the small black cross does not appear when hovering over the corner of the cell, the option might be disabled in the Excel settings. To fix this, navigate to **File > Options > Advanced**, and under the Editing Options section, ensure the checkbox labeled "Enable fill handle and cell drag-and-drop" is checked.

**Issue: Formulas are not updating row numbers (Absolute Reference).** If, upon dragging, every cell shows the exact same value (e.g., `=Sheet1!B2` for every cell), it means the initial formula was created using absolute cell reference syntax (e.g., `=Sheet1!$B$2`). For vertical autofill, ensure only relative references are used for the row component (e.g., `=Sheet1!B2`).

**Issue: Double-Click Fails.** The double-click function of the Fill Handle relies on a contiguous column of data immediately adjacent to the target column (in our example, Column B or Column A in Sheet2). If Column B contains blank cells, the autofill will stop prematurely at the first blank cell encountered. In such cases, the manual click-and-drag method must be used to cover the full required range.

## Alternative Methods for Data Transfer: Lookup Functions

While the direct autofill of relative references is perfect when the source and destination data are aligned row-for-row, data integration often requires matching data based on a unique identifier, especially when the rows are in a different order. In such scenarios, advanced lookup functions are necessary.

The most common alternatives include **VLOOKUP** (Vertical Lookup) and the more flexible combination of **INDEX and MATCH**, or its modern successor, **XLOOKUP**. These functions search for a specific value (the lookup value, e.g., a student's name) in one column of the source sheet

and return a corresponding value from another column (e.g., their score).

When implementing VLOOKUP, the formula is generally structured as `=VLOOKUP(lookup_value, table_array, col_index_num, )`. The `table_array` component would include the cross-sheet reference (e.g., `Sheet1!A:B`). Even when using these complex lookup functions, the Fill Handle is still the essential tool used to drag the initial lookup formula down the column, allowing Excel to adjust the lookup value and thus completing the data retrieval for all rows efficiently.

Therefore, regardless of whether you use a simple `=Sheet1!B2` reference or a complex VLOOKUP formula, mastering the quick application of the Fill Handle remains the core productivity technique for scaling formulas across an entire column in Excel.

## Summary of Cross-Sheet Autofill Best Practices

To summarize the most effective way to utilize the autofill feature for cross-sheet data transfer, adherence to specific best practices is recommended:

**Verify Alignment:** Ensure that the desired data transfer is truly row-for-row. If the rows are sorted differently, use lookup functions instead of direct cell references.

**Start with Relative References:** The initial formula (e.g., `=Sheet1!B2`) must use relative references unless you specifically intend to pull the same single value into every cell in the destination column.

**Leverage the Double-Click:** Whenever possible, use the double-click functionality on the Fill Handle for rapid deployment, ensuring the adjacent columns are contiguous.

**Check for Consistency:** After autofilling, spot-check a few cells at the bottom of the column to confirm that the cell reference (e.g., `=Sheet1!B100`) adjusted correctly.

By integrating these steps, you can harness the full power of Excel's automation features, ensuring that your workbooks are dynamically linked, accurate, and scalable for any size of dataset.