

How to Easily Calculate Standard Deviation in Excel with VBA

Authored by
stats writer

November 20, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Calculate Standard Deviation in Excel with VBA*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=98250>

The calculation of the standard deviation (STDEV) is a fundamental task in statistical analysis, providing insight into the dispersion of data points within a dataset. While Microsoft Excel offers built-in worksheet functions for this purpose, leveraging VBA allows for greater automation and integration into complex workflows. Calculating the standard deviation of a defined range using VBA requires setting up a procedure that calls Excel's statistical functions and then outputs the result, either directly into a cell or via an informational dialog box.

This process typically involves referencing the specific range of interest using a VBA macro, utilizing the powerful WorksheetFunction.StDev method to perform the required mathematical calculation, and finally, using display methods like assigning the value to a cell or employing the MsgBox function to present the computed standard deviation to the user. This approach ensures robust and repeatable statistical operations directly within the Excel environment.

Core VBA Syntax for Standard Deviation Calculation

When automating statistical calculations in VBA, the most efficient method is to access Excel's native functions through the WorksheetFunction object. This object grants VBA direct access to the vast library of functions available in the Excel worksheet environment, ensuring consistency and reliability in statistical output. For calculating the sample standard deviation, the StDev function is employed.

The basic structure for calculating the standard deviation and placing the output directly into a worksheet cell is straightforward. We define a subroutine and then use the Range object to specify the destination cell, equating it to the result of the WorksheetFunction.StDev applied to the target data range. This method is ideal when you need the resulting value to be a static part of your spreadsheet report or dashboard.

You can use the following basic syntax to calculate the standard deviation of values in a range using VBA, assigning the output directly to a cell:

```
Sub StDevRange()  
Range("D2") = WorksheetFunction.StDev(Range("B2:B11"))  
End Sub
```

This particular example demonstrates the calculation of the standard deviation for all numerical values found in the range **B2:B11**. The resulting calculated statistic is then immediately assigned and displayed within cell **D2** of the active worksheet. This minimizes the need for manual formula entry and allows for quick, automated reporting.

Alternative Method: Displaying Results via Message Box

While assigning the result to a cell is useful for permanent recording, often a user only needs a momentary display of the calculated statistic for verification or immediate feedback. In these scenarios, the `MsgBox` function provides a clean, non-intrusive way to present the output. This approach requires declaring a variable to temporarily hold the calculated value before displaying it.

To implement this, we first dimension a variable (e.g., `stdev`) as a Single or Double data type, suitable for storing floating-point numerical results. We then perform the calculation using `WorksheetFunction.StDev`, storing the output in this declared variable. Finally, the `MsgBox` function is called, concatenating a descriptive string with the variable holding the calculated standard deviation. This provides a clear, labeled result to the user.

If you prefer to display the standard deviation of values within a modal message box instead of writing it to a cell, you can use the following revised syntax:

Sub StDevRange()

'Create variable to store standard deviation of values

Dim stdev As Single

'Calculate standard deviation of values in range

stdev = WorksheetFunction.StDev(Range("B2:B11"))

'Display the result

MsgBox "Standard Deviation of Values in Range: " & stdev

End Sub

Setting the Stage: The Example Dataset

To fully illustrate the utility and execution of these VBA methods, we will apply them to a practical dataset. The following examples utilize a sample sheet containing performance metrics for various basketball players. Specifically, we are interested in analyzing the dispersion of the 'Points' scored by these players, located in column B of the spreadsheet.

Understanding the context of the data is crucial before calculating the standard deviation. This statistic will help determine how much the individual player scores deviate from the average score of the group. A high standard deviation would indicate a wide spread of scores (some players scoring very high and others very low), while a low standard deviation suggests consistency among the players.

The visual representation below shows the data used for subsequent calculations. The target data

range for the standard deviation calculation is highlighted implicitly as the values in column B (Points), extending from row 2 to row 11.

	A	B	C	D	E	F
1	Team	Points				
2	Mavs	22				
3	Heat	20				
4	Spurs	40				
5	Rockets	43				
6	Nets	39				
7	Warriors	24				
8	Thunder	10				
9	Hawks	13				
10	Magic	19				
11	Kings	15				
12						
13						
14						
15						
16						
17						
18						
19						

Example 1: Displaying Standard Deviation Directly in a Worksheet Cell

In our first practical demonstration, we will implement the method that writes the statistical result directly back into the Excel worksheet. Suppose the objective is to calculate the standard deviation of the 'Points' column (Range B2:B11) and output the resulting value into cell D2, making the result easily accessible alongside the data for reporting purposes.

To achieve this, we create a simple macro named `StDevRange`. Within this subroutine, we use the `WorksheetFunction.StDev` method, specifying the precise range B2:B11 as the input argument. We then instruct VBA to place the output into the target cell, D2.

We can utilize the following VBA code structure to execute this task:

```
Sub StDevRange()  
Range("D2") = WorksheetFunction.StDev(Range("B2:B11"))  
End Sub
```

End Sub

Upon execution of this macro, the VBA code runs silently in the background, retrieving the data points from B2 through B11, calculating the sample standard deviation, and immediately displaying the computed value in cell D2 of the active spreadsheet. This automated approach bypasses the need for manual formula entry (e.g., `=STDEV.S(B2:B11)`) and is highly efficient for recurring tasks.

The visual outcome after running the macro confirms the successful calculation and placement of the result:

	A	B	C	D	E	F
1	Team	Points				
2	Mavs	22		11.93734		
3	Heat	20				
4	Spurs	40				
5	Rockets	43				
6	Nets	39				
7	Warriors	24				
8	Thunder	10				
9	Hawks	13				
10	Magic	19				
11	Kings	15				
12						
13						
14						
15						
16						
17						
18						

A closer inspection reveals that cell **D2** now contains the value **11.93734**. This value represents the standard deviation for the points scored by the basketball players in our dataset. It quantifies the typical amount of variation from the mean score.

Example 2: Presenting Standard Deviation Results in a Message Box

For scenarios where the calculated statistic does not need to be permanently stored on the sheet but requires immediate viewing, leveraging the [MsgBox function](#) within [VBA](#) is the preferred

method. This approach offers a clean, transient pop-up that draws the user's attention to the result without modifying the worksheet layout, outside of the calculation itself.

As previously discussed, this method requires defining an intermediate variable to store the output of the `WorksheetFunction.StDev`. Using the `Dim` statement ensures memory is allocated for the numeric result, allowing it to be dynamically passed to the `MsgBox` function.

If we want to calculate the standard deviation of the values in the 'Points' column (B2:B11) and display it in a prompt, the following robust macro should be used:

```
Sub StDevRange()
```

```
'Create variable to store standard deviation of values
```

```
Dim stdev As Single
```

```
'Calculate standard deviation of values in range
```

```
stdev = WorksheetFunction.StDev(Range("B2:B11"))
```

```
'Display the result
```


```
MsgBox "Standard Deviation of Values in Range: " & stdev
```

```
End Sub
```

Executing this code will pause the Excel session temporarily and display a message box containing the calculated statistic, providing the user with immediate feedback on the data variability.

After running this revised macro, the user is presented with the following output:

	A	B	C	D	E	F	G	H
1	Team	Points						
2	Mavs	22						
3	Heat	20						
4	Spurs	40						
5	Rockets	43						
6	Nets	39						
7	Warriors	24						
8	Thunder	10						
9	Hawks	13						
10	Magic	19						
11	Kings	15						
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								



Microsoft Excel

Standard Deviation of Values in Range: 11.9373363863133

OK

The message box clearly states that the standard deviation of the values within the range **B2:B11** is approximately **11.937** (often rounded for display purposes). This method is highly effective for auditing or debugging purposes where quick numerical checks are necessary.

Flexible Range Selection: Targeting Specific Ranges vs. Entire Columns

The examples provided above focused on calculating the standard deviation for a defined, limited range: **B2:B11**. This specificity is essential when working with structured tables or datasets where the footer rows or empty cells below the data must be excluded from the statistical calculation to maintain accuracy.

However, VBA offers flexibility in defining the range argument for the `WorksheetFunction.StDev`. If the dataset spans the entire column, or if you prefer a dynamic calculation that always includes all non-blank cells in a given column, you can reference the entire column using the syntax `"B:B"` instead of specifying row numbers (e.g., `"B2:B11"`).

When the range is set to `Range("B:B")`, the StDev function will automatically evaluate every cell within column B, excluding non-numeric values (such as the header text). This comprehensive

selection is helpful when dealing with constantly growing datasets where new rows are appended regularly.

It is important to note that while calculating the standard deviation of an entire column (e.g., B:B) is possible, analysts should exercise caution. Including header rows, unnecessary empty cells, or unrelated footers in the calculation can skew the resulting standard deviation, potentially leading to inaccurate statistical conclusions about the core dataset.

For detailed technical specifications and advanced implementation options related to the standard deviation calculation within VBA, users are strongly advised to consult the official Microsoft documentation.

You can use the following basic syntax to calculate the of values in a range using VBA:

```
Sub StDevRange()  
Range("D2") = WorksheetFunction.StDev(Range("B2:B11"))  
End Sub
```

This particular example calculates the standard deviation of values in the range **B2:B11** and assigns the result to cell **D2**.

If you would instead like to display the standard deviation of values in a message box, you can use the following syntax:

```
Sub StDevRange()  
'Create variable to store standard deviation of values  
Dim stdev As Single  
  
'Calculate standard deviation of values in range  
stdev = WorksheetFunction.StDev(Range("B2:B11"))  
  
'Display the result  
MsgBox "Standard Deviation of Values in Range: " & stdev  
End Sub
```

The following examples shows how to use each of these methods in practice with the following dataset in Excel that contains information about various basketball players:

	A	B	C	D	E	F
1	Team	Points				
2	Mavs	22				
3	Heat	20				
4	Spurs	40				
5	Rockets	43				
6	Nets	39				
7	Warriors	24				
8	Thunder	10				
9	Hawks	13				
10	Magic	19				
11	Kings	15				
12						
13						
14						
15						
16						
17						
18						
19						

Example 1: Calculate Standard Deviation of Range Using VBA and Display Results in Cell

Suppose we would like to calculate the standard deviation of values in the points column and output the results in a specific cell.

We can create the following macro to do so:

```
Sub StDevRange()  
Range("D2") = WorksheetFunction.StDev(Range("B2:B11"))  
End Sub
```

When we run this macro, we receive the following output:

	A	B	C	D	E	F
1	Team	Points				
2	Mavs	22		11.93734		
3	Heat	20				
4	Spurs	40				
5	Rockets	43				
6	Nets	39				
7	Warriors	24				
8	Thunder	10				
9	Hawks	13				
10	Magic	19				
11	Kings	15				
12						
13						
14						
15						
16						
17						
18						

Notice that cell **D2** contains a value of **11.93734**.

This tells us that the standard deviation of values in the points column is 11.93734.

Example 2: Calculate Standard Deviation of Range Using VBA and Display Results in Message Box

Suppose we would instead like to calculate the standard deviation of values in the points column and output the results in a message box.

Sub StDevRange()

'Create variable to store standard deviation of values

Dim stdev As Single

'Calculate standard deviation of values in range

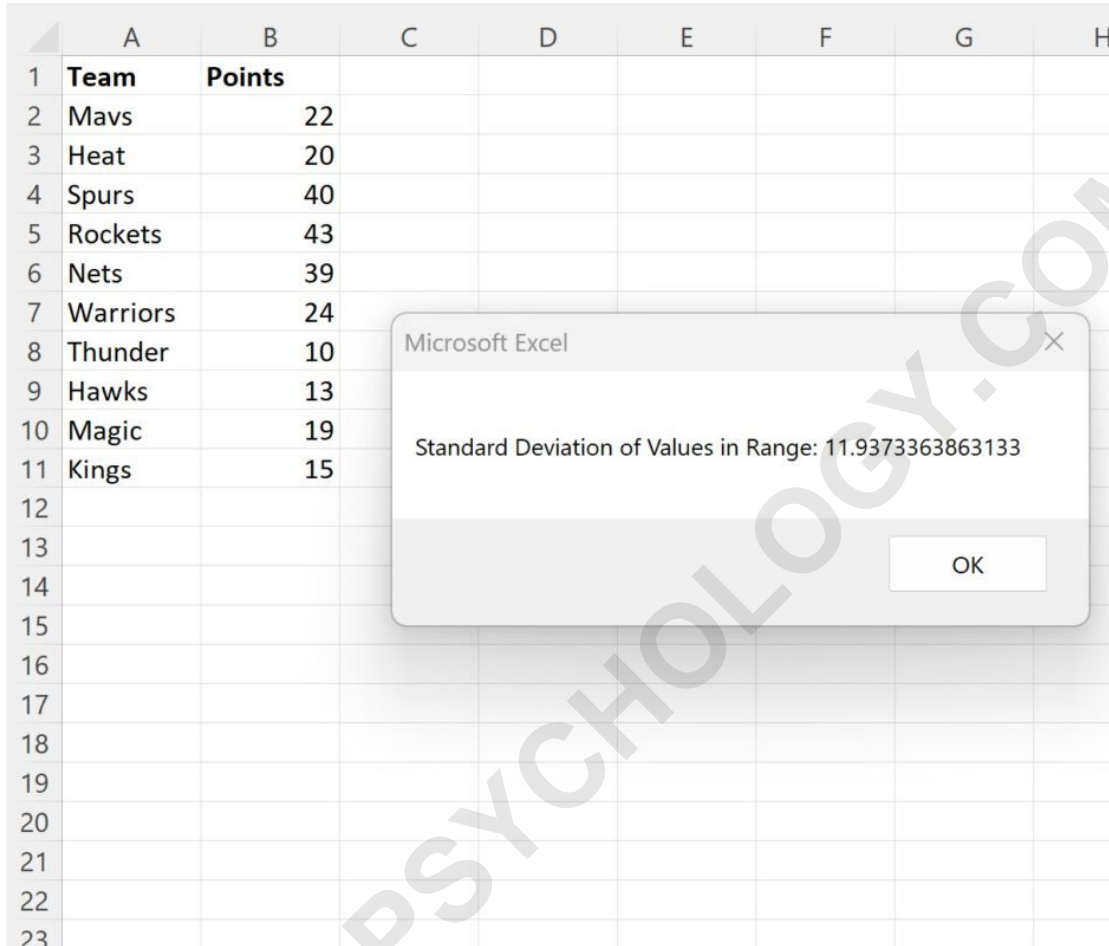
stdev = WorksheetFunction.StDev(Range("B2:B11"))

'Display the result

MsgBox "Standard Deviation of Values in Range: " & stdev

End Sub

When we run this macro, we receive the following output:



The screenshot shows an Excel spreadsheet with two columns: 'Team' and 'Points'. The data is as follows:

	A	B	C	D	E	F	G	H
1	Team	Points						
2	Mavs	22						
3	Heat	20						
4	Spurs	40						
5	Rockets	43						
6	Nets	39						
7	Warriors	24						
8	Thunder	10						
9	Hawks	13						
10	Magic	19						
11	Kings	15						
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								

Overlaid on the spreadsheet is a Microsoft Excel message box with the following text:

Standard Deviation of Values in Range: 11.9373363863133

OK

The message box tells us that the standard deviation of values in the range **B2:B11** is **11.937**.

Note that in this example we calculated the standard deviation of values in the range **B2:B11**.

However, if you'd like to instead calculate the standard deviation of values in an entire column you could type **B:B** instead.

This will calculate the standard deviation of values for every numeric cell in column B.

Note: You can find the complete documentation for the VBA **StDev** method [here](#).

Conclusion and Next Steps

Using VBA provides a powerful and flexible solution for integrating statistical analysis directly into

your Excel workflows. Whether you choose to display the standard deviation result in a specific worksheet cell for permanent record-keeping or use a temporary MsgBox function for quick analysis, the `WorksheetFunction.StDev` method remains the cornerstone of the calculation.

Mastering these basic VBA structures allows for the development of more complex statistical macros, such as calculating standard deviation only for filtered data, looping through multiple ranges, or integrating the result into larger automated reporting systems. Always ensure that the specified range accurately reflects the data you intend to analyze to maintain the integrity of your statistical outcomes.

ARABPSYCHOLOGY.COM