How can I rank a list of values using VBA?

Authored by

stats writer

June 24, 2024

VBA (Visual Basic for Applications) is a programming language commonly used in Microsoft Excel to automate tasks and perform calculations. In order to rank a list of values using VBA, one can use the built-in function "RANK" which calculates the rank of a value in a given data set. This function takes two arguments, the first being the value to be ranked and the second being the range of values to be ranked against. The function then returns the rank of the value within the range, with the highest value being ranked as 1. By using a loop and the "RANK" function, one can easily rank a list of values in ascending or descending order, depending on the desired outcome. This allows for efficient and accurate organization of data, making it a useful tool for data analysis and decision making.

VBA: Rank a List of Values

You can use the following basic syntax to rank a list of values in Excel using VBA:

```
Sub RankValues()
Dim i As Integer

For i = 2 To 11
Range("C" & i) = WorksheetFunction.Rank(Range("B" & i), Range("B2:B11"), 0)
Next i
End Sub
```

This particular example ranks the values in cells B2:B11 and outputs the ranks in cells C2:C11.

The last argument of of 0 specifies that we should rank

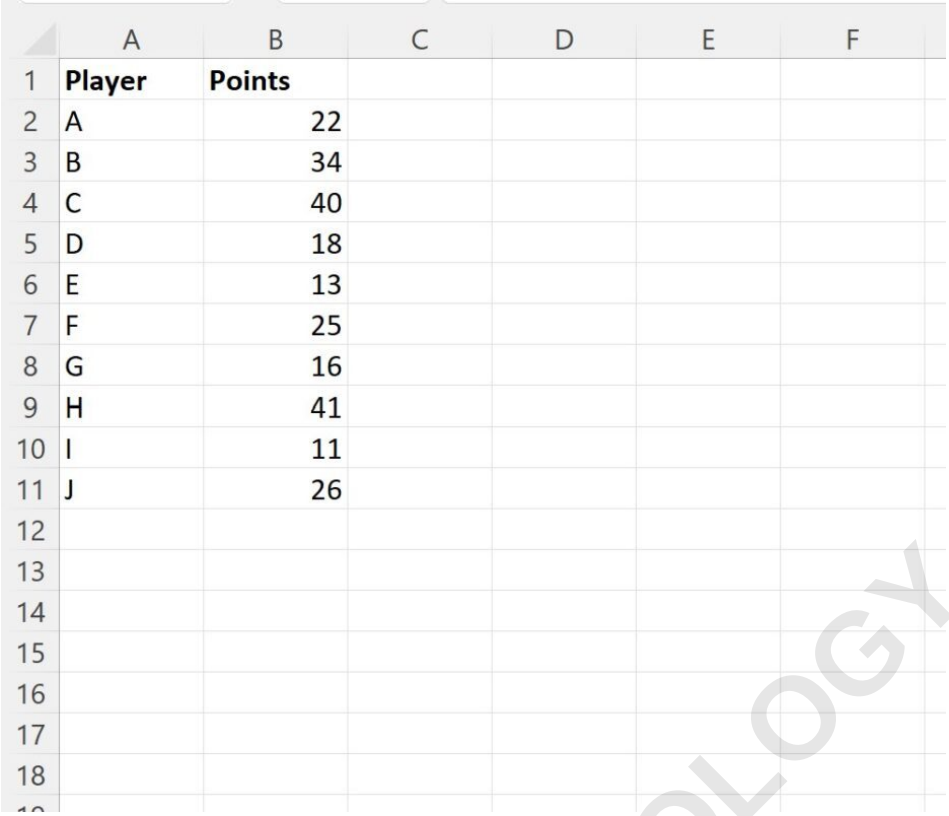
the values in ascending order (the largest value receives a rank of 1, the second largest value receives a rank of 2, etc.).

To rank the values in descending order, simply change the 0 to 1.

The following example shows how to use this syntax in practice.

Example: How to Rank Values Using VBA

Suppose we have the following list of basketball players along with their points scored:



	A	B	C	D	E	F
1	**Player**	**Points**				
2	A	22				
3	B	34				
4	C	40				
5	D	18				
6	E	13				
7	F	25				
8	G	16				
9	H	41				
10	I	11				
11	J	26				
12						
13						
14						
15						
16						
17						
18						

Suppose we would like to calculate the rank of each value in the points column.

We can create the following macro to do so:

```
Sub RankValues()
Dim i As Integer

For i = 2 To 11
Range("C" & i) = WorksheetFunction.Rank(Range("B" & i), Range("B2:B11"), 0)
Next i
```

End Sub

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

	A	B	C	D	E	F
1	**Player**	**Points**				
2	A	22	6			
3	B	34	3			
4	C	40	2			
5	D	18	7			
6	E	13	9			
7	F	25	5			
8	G	16	8			
9	H	41	1			
10	I	11	10			
11	J	26	4			
12						
13						
14						
15						
16						
17						
18						
19						

The rank of each value in the points column is displayed in column C.

For example:

Player H with 41 points has the highest points value, so they receive a rank of 1.Player C with 40 points has the

second highest points value, so they receive a rank of 2.

To instead rank the values in the points column in descending order, we can change the last argument in the Rank method from 0 to 1:

```
Sub RankValues()
Dim i As Integer

For i = 2 To 11
Range("C" & i) = WorksheetFunction.Rank(Range("B" & i), Range("B2:B11"), 1)
Next i
End Sub
```

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

	A	B	C	D	E	F
1	**Player**	**Points**				
2	A	22	5			
3	B	34	8			
4	C	40	9			
5	D	18	4			
6	E	13	2			
7	F	25	6			
8	G	16	3			
9	H	41	10			
10	I	11	1			
11	J	26	7			
12						
13						
14						
15						
16						
17						
18						

The rank of each value in the points column is displayed in column C.

For example:

Player I with 11 points has the lowest points value, so they receive a rank of 1.Player E with 13 points has the second lowest points value, so they receive a rank of 2.

And so on.

Note: You can find the complete documentation for the

VBA Rank method .