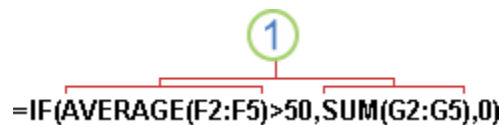


Using Functions to Create, Convert and Interpret Data

About nested functions(functions within functions)

In certain cases, you may need to use a function as one of the arguments (argument: The values that a function uses to perform operations or calculations. The type of argument a function uses is specific to the function. Common arguments that are used within functions include numbers, text, cell references, and names.) of another function. For example, the following formula uses a nested AVERAGE function and compares the result with the value 50.



=IF(AVERAGE(F2:F5)>50,SUM(G2:G5),0)

Valid returns When a nested function is used as an argument, it must return the same type of value that the argument uses. For example, if the argument returns a TRUE or FALSE value, then the nested function must return a TRUE or FALSE. If it doesn't, Microsoft Excel displays a **#VALUE!** error value.

Nesting level limits A formula can contain up to seven levels of nested functions. When Function B is used as an argument in Function A, Function B is a second-level function. For instance, the AVERAGE function and the SUM function are both second-level functions because they are arguments of the IF function. A function nested within the AVERAGE function would be a third-level function, and so on.

VLOOKUP

Searches for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify in the table. Use VLOOKUP instead of HLOOKUP when your comparison values are located in a column to the left of the data you want to find.

The V in VLOOKUP stands for "Vertical."

Syntax

VLOOKUP(lookup_value,table_array,col_index_num,range_lookup)

Lookup_value is the value to be found in the first column of the array. Lookup_value can be a value, a reference, or a text string.

Table_array is the table of information in which data is looked up. Use a reference to a range or a range name, such as Database or List.

- If `range_lookup` is `TRUE`, the values in the first column of `table_array` must be placed in ascending order: ..., -2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE; otherwise `VLOOKUP` may not give the correct value. If `range_lookup` is `FALSE`, `table_array` does not need to be sorted.
- You can put the values in ascending order by choosing the **Sort** command from the **Data** menu and selecting **Ascending**.
- The values in the first column of `table_array` can be text, numbers, or logical values.
- Uppercase and lowercase text are equivalent.

`Col_index_num` is the column number in `table_array` from which the matching value must be returned. A `col_index_num` of 1 returns the value in the first column in `table_array`; a `col_index_num` of 2 returns the value in the second column in `table_array`, and so on. If `col_index_num` is less than 1, `VLOOKUP` returns the #VALUE! error value; if `col_index_num` is greater than the number of columns in `table_array`, `VLOOKUP` returns the #REF! error value.

`Range_lookup` is a logical value that specifies whether you want `VLOOKUP` to find an exact match or an approximate match. If `TRUE` or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than `lookup_value` is returned. If `FALSE`, `VLOOKUP` will find an exact match. If one is not found, the error value #N/A is returned.

Remarks

- If `VLOOKUP` can't find `lookup_value`, and `range_lookup` is `TRUE`, it uses the largest value that is less than or equal to `lookup_value`.
- If `lookup_value` is smaller than the smallest value in the first column of `table_array`, `VLOOKUP` returns the #N/A error value.
- If `VLOOKUP` can't find `lookup_value`, and `range_lookup` is `FALSE`, `VLOOKUP` returns the #N/A value.

FREQUENCY

Calculates how often values occur within a range of values, and then returns a vertical array of numbers. For example, use `FREQUENCY` to count the number of test scores that fall within ranges of scores. Because `FREQUENCY` returns an array, it must be entered as an array formula.

Syntax

FREQUENCY(data_array,bins_array)

`Data_array` is an array of or reference to a set of values for which you want to count frequencies. If `data_array` contains no values, `FREQUENCY` returns an array of zeros.

`Bins_array` is an array of or reference to intervals into which you want to group the values in `data_array`. If `bins_array` contains no values, `FREQUENCY` returns the number of elements in `data_array`.

Remarks

- `FREQUENCY` is entered as an array formula after you select a range of adjacent cells into which you want the returned distribution to appear.
- The number of elements in the returned array is one more than the number of elements in `bins_array`. The extra element in the returned array returns the count of any values above the highest interval. For example, when counting three ranges of values (intervals) that are entered into three cells, be sure to enter `FREQUENCY` into four cells for the results. The extra cell returns the number of values in `data_array` that are greater than the third interval value.
- `FREQUENCY` ignores blank cells and text.
- Formulas that return arrays must be entered as array formulas.

Example

This example assumes all test scores are integers.

The example may be easier to understand if you copy it to a blank worksheet.

	A		B
	Scores		Bins
1			
2	79		70
3	85		79
4	78		89
5	85		
6	50		
7	81		
8	95		
9	88		
10	97		
	Formula		Description (Result)
	<code>=FREQUENCY(A2:A10,B2:B5)</code>		Number of scores less than or equal to 70 (1)
			Number of scores in the bin 71-79 (2)
			Number of scores in the bin 80-89 (4)
			Number of scores greater than or equal to 90 (2)

SUMIF

Adds the cells specified by a given criteria.

Syntax

SUMIF(range,criteria,sum_range)

Range is the range of cells you want evaluated.

Criteria is the criteria in the form of a number, expression, or text that defines which cells will be added. For example, criteria can be expressed as 32, "32", ">32", "apples".

Sum_range are the actual cells to sum.

Remarks

- The cells in sum_range are summed only if their corresponding cells in range match the criteria.
- If sum_range is omitted, the cells in range are summed.
- Microsoft Excel provides additional functions that can be used to analyze your data based on a condition. For example, to count the number of occurrences of a string of text or a number within a range of cells, use the COUNTIF function. To have a formula return one of two values based on a condition, such as a sales bonus based on a specified sales amount, use the IF function.

Example

The example may be easier to understand if you copy it to a blank worksheet.

	A	B
1	Property Value	Commission
2	100	7
3	200	14
4	300	21
5	400	28

Formula	Description (Result)
=SUMIF(A2:A5,">160000",B2:B5)	Sum of the commissions for property values over 160000 (63,000)

COUNTIF

Counts the number of cells within a range that meet the given criteria.

Syntax

COUNTIF(range,criteria)

Range is the range of cells from which you want to count cells.

Criteria is the criteria in the form of a number, expression, or text that defines which cells will be counted. For example, criteria can be expressed as 32, "32", ">32", "apples".

Remark

Microsoft Excel provides additional functions that can be used to analyze your data based on a condition. For example, to calculate a sum based on a string of text or a number within a range, use the SUMIF worksheet function. To have a formula return one of two values based on a condition, such as a sales bonus based on a specified sales amount, use the IF worksheet function.

Example

The example may be easier to understand if you copy it to a blank worksheet.

	A	B
1	Data	Data
2	apples	32
3	oranges	54
4	peaches	75
5	apples	86

Formula	Description (Result)
=COUNTIF(A2:A5,"apples")	Number of cells with apples in the first column above (2)
=COUNTIF(B2:B5,">55")	Number of cells with a value greater than 55 in the second column above (2)