Formula Tips in Excel
Part 2 - 2010/2016

Though it is advisable to keep the number of formulas in use to a minimum, formulas are a necessary component of most Excel workbooks. If you know which functions you should insert into formulas to achieve specific results, you are far more likely to be successful when working with Excel. Further, if you are familiar with some of the fundamentals of formulas such as absolute and relative cell referencing, you will help to minimize the likelihood of errors in your workbooks. In this webinar, we will explore these and other formula fundamentals.

Looking Up Data with VLOOKUP

One of the advanced features of Excel is its ability to look up items from tables of data using lookup functions. The most commonly used lookup function is **VLOOKUP**, or vertical lookup, which is used to look up items that are arranged vertically in a table. Accounting professionals will find this function useful in retrieving data from tax or inventory tables. The syntax for the VLOOKUP function is:

VLOOKUP(lookup value, table array, column index, [True/False])

where:

lookup value is the value to be looked up in the table,
table array is the table in which the value is to be looked up,
column index is the column in the table from which the data is returned, and
True/False is an optional argument. If the option is set to *False*, the first column in the table array need not be sorted in ascending order, and the function will only find an *exact match* of the lookup value. If an exact match does not exist, the function returns #NA! If the option is set to *True* or left out of the function, the first column in the table array *must* be sorted in ascending order, and the function may find an *approximate match*, defined as the largest value that is less than or equal to the lookup value.

In our first example, we will compute corporate income tax using data stored in a tax table that serves as the table array for several VLOOKUP functions. The formula uses three VLOOKUP functions – one to look up the base tax, one to help calculate the amount of income above the base tax bracket, and one to look up the tax rate on income above the base tax bracket as shown in **Figure 1**.

Note that the formulas in this example use cell references. An alternative, and perhaps better, approach would have been to use defined names in lieu of cell references.

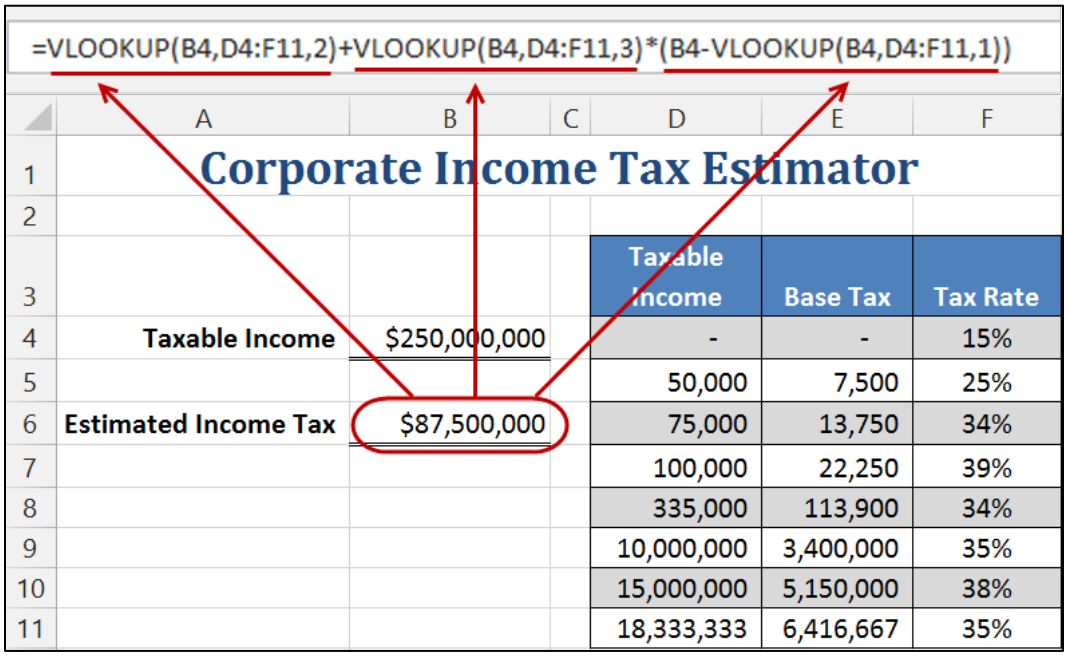


Figure 1 - Using VLOOKUP to Calculate Corporate Income Tax

Our second example uses VLOOKUP to return information related to an inventory list maintained in Excel. In this case, users enter an item number, and VLOOKUP functions return the item name, price, and quantity on hand. Note that the optional **True/False** argument is set to **FALSE** so that VLOOKUP requires an *exact match*. We certainly don't want the function to return an *approximate match* if the item is not found in inventory! **Figure 117** contains two sets of VLOOKUP functions, one with the **False** argument set and the other without the optional argument. Note that when the False argument is set, VLOOKUP returns **#N/A** if an exact match is not found, but without the argument, it returns an approximate match *even when the item is not in the inventory list*. The reason for this is because, in default, the VLOOKUP function finds and returns information related to the

largest value found that is *less than or equal to* the lookup value. If the False argument had not been set in our formula, VLOOKUP would have returned information about Item **C003**, the largest value that is less than or equal to Item **C004**, the lookup value. Hence, the information returned would be in error and could mislead users. Whenever an exact match is required, make sure to set the optional False argument.

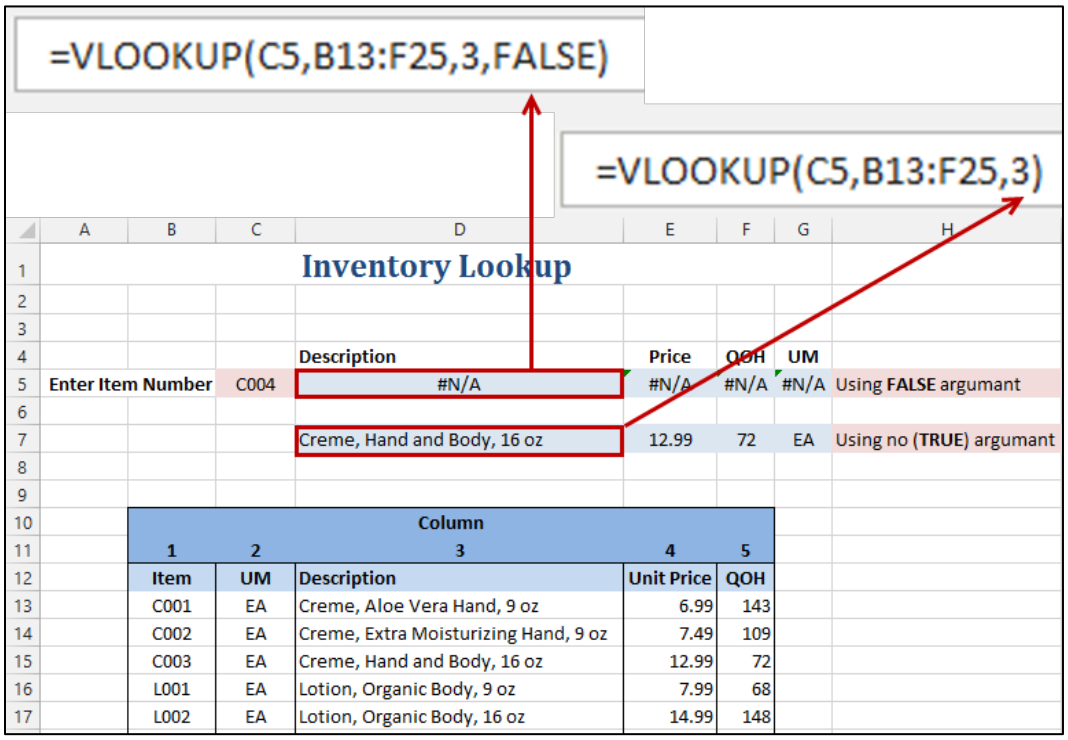


Figure 2 - Using FALSE Argument to Require an Exact Match

Looking Up Data with HLOOKUP

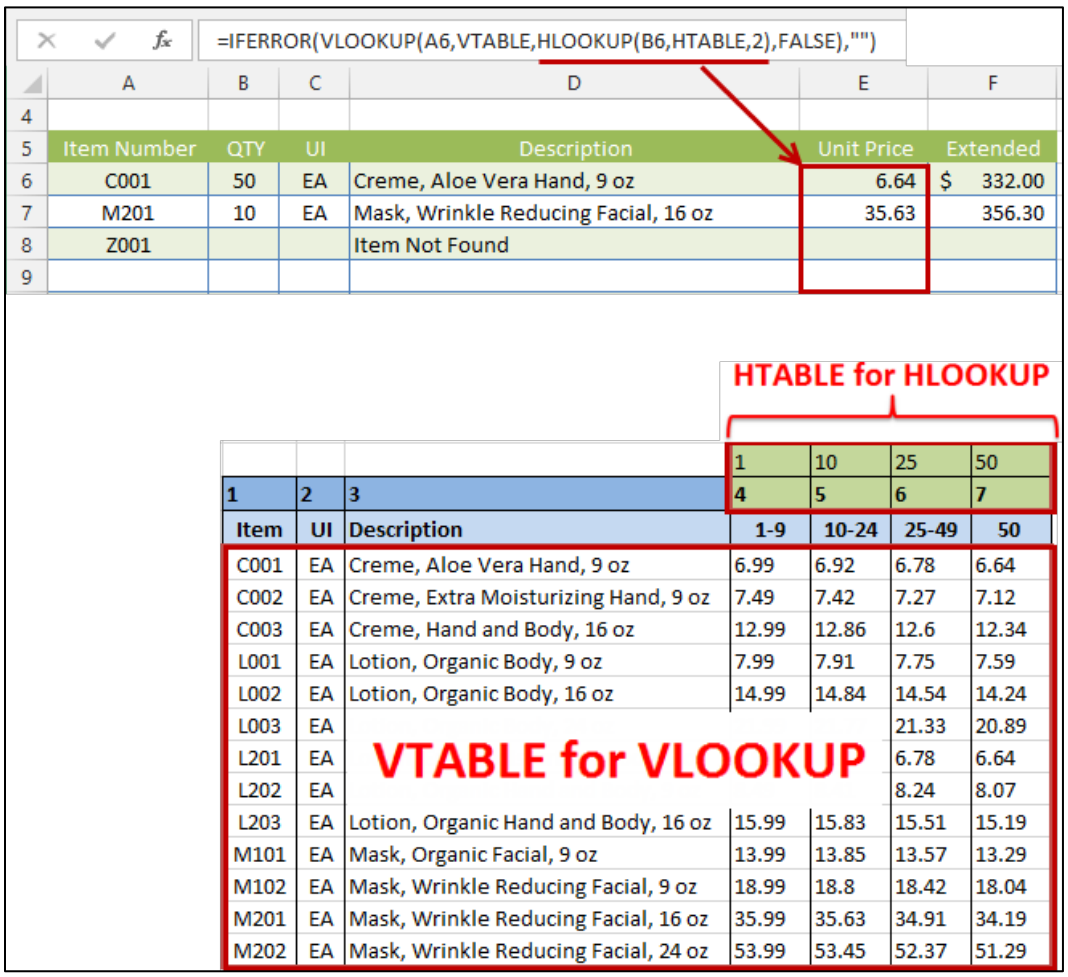
Just as the VLOOKUP function looks up values in columns, **HLOOKUP** looks up values in rows. The syntax for the HLOOKUP function is:

HLOOKUP(lookup value, table array, row index, [True/False])

where:

lookup value is the value to be looked up in the table,
table array is the table in which the value is to be looked up,
row index is the row in the table from which the data is returned, and
True/False is an optional argument. If the option is set to *False*, the first row in the table array need not be sorted in ascending order, and the function will only find an *exact match* of the lookup value. If an exact match does not exist, the function returns #NA! If the option is set to *True* or left out of the function, the first row in the table array *must* be sorted in ascending order (left to right), and the function may find an *approximate match*, defined as the largest value that is less than or equal to the lookup value.

The following example uses the HLOOKUP function as an argument in a VLOOKUP formula that returns the price of an item depending upon the quantity ordered by a customer. Customers are given price breaks based on the number of units purchased on an item-by-item basis. Individual sales price levels are set for the following ranges: zero to nine items, ten to twenty-four items, twenty-five to forty-nine items, and fifty items or more.



Item Number	QTY	UI	Description	Unit Price	Extended
C001	50	EA	Creme, Aloe Vera Hand, 9 oz	6.64	$ 332.00
M201	10	EA	Mask, Wrinkle Reducing Facial, 16 oz	35.63	356.30
Z001			Item Not Found		

			1	10	25	50
1	2	3	4	5	6	7
Item	UI	Description	1-9	10-24	25-49	50
C001	EA	Creme, Aloe Vera Hand, 9 oz	6.99	6.92	6.78	6.64
C002	EA	Creme, Extra Moisturizing Hand, 9 oz	7.49	7.42	7.27	7.12
C003	EA	Creme, Hand and Body, 16 oz	12.99	12.86	12.6	12.34
L001	EA	Lotion, Organic Body, 9 oz	7.99	7.91	7.75	7.59
L002	EA	Lotion, Organic Body, 16 oz	14.99	14.84	14.54	14.24
L003	EA				21.33	20.89
L201	EA				6.78	6.64
L202	EA				8.24	8.07
L203	EA	Lotion, Organic Hand and Body, 16 oz	15.99	15.83	15.51	15.19
M101	EA	Mask, Organic Facial, 9 oz	13.99	13.85	13.57	13.29
M102	EA	Mask, Wrinkle Reducing Facial, 9 oz	18.99	18.8	18.42	18.04
M201	EA	Mask, Wrinkle Reducing Facial, 16 oz	35.99	35.63	34.91	34.19
M202	EA	Mask, Wrinkle Reducing Facial, 24 oz	53.99	53.45	52.37	51.29

Figure 3 - Using HLOOKUP to Specify the Data Column in VLOOKUP

In this case, the HLOOKUP function supplies the third argument – the column index – of the VLOOKUP function as shown in **Figure 3**. In ascending order from left to right in the first row of HTABLE, the defined name that references the horizontal table array is the values 1, 10, 25, and 50, which coincide with the minimum values of the sales price levels. The second row contains the column numbers – 4, 5, 6, and 7 – from which to retrieve the item prices in the VLOOKUP table array. For example, if the quantity entered was 23, the HLOOKUP function would return 5, the column from which to retrieve the price for the item selected.

Trapping Errors with IFERROR

The **IFERROR** function has the potential to simplify creating complex formulas because it traps all errors with a single function. In the past, users had to employ unintuitive formulas to trap simple errors or use multiple error-trapping functions, each of which trapped very specific errors. The syntax for IFERROR is as follows.

IFERROR(original value or formula, value if error)

Continuing from the previous example, note that when the **False** argument is set in a VLOOKUP function, the function returns **#N/A** when an exact match is not found. However, the formulas can be modified to return "Item Not Found" in the Description column and blank cells in the **Price**, **QOH**, and **UM** columns. To modify the item description formula, incorporate the IFERROR function as shown in **Figure 4**. For the price, quantity on hand, and unit of measure formulas, use IFERROR to return null strings ("").

```
=IFERROR(VLOOKUP(C5,B13:F25,3,FALSE), "Item Not Found")
```

Figure 4 - Using IFERROR with VLOOKUP to Trap Errors

The result of using IFERROR along with VLOOKUP is shown in **Figure 5**, along with a formula that returns an

approximate match for comparative purposes.

	A	B	C	D	E	F	G
4				**Description**	**Price**	**QOH**	**UM**
5	**Enter Item Number**		C004	Item Not Found			
6							
7				Creme, Hand and Body, 16 oz	12.99	72	EA

Figure 5 - Results of Using IFERROR and VLOOKUP Together

Excel has several error-identifying functions. Among them are **ISERR**, **ISERROR**, and **ISNA**. When these functions are used to trap errors, they must be combined with the IF function to provide the necessary logic on what to do if an error is encountered. The IFERROR function, on the other hand, combines error trapping with the necessary logic in a single, easy-to-use function.

Conditional and Boolean Calculations

The **IF** function is used to make calculations based on whether specified conditions are met. The two latest versions of Excel support up to sixty-four (64) nested IFs in each formula. The syntax of the IF function follows.

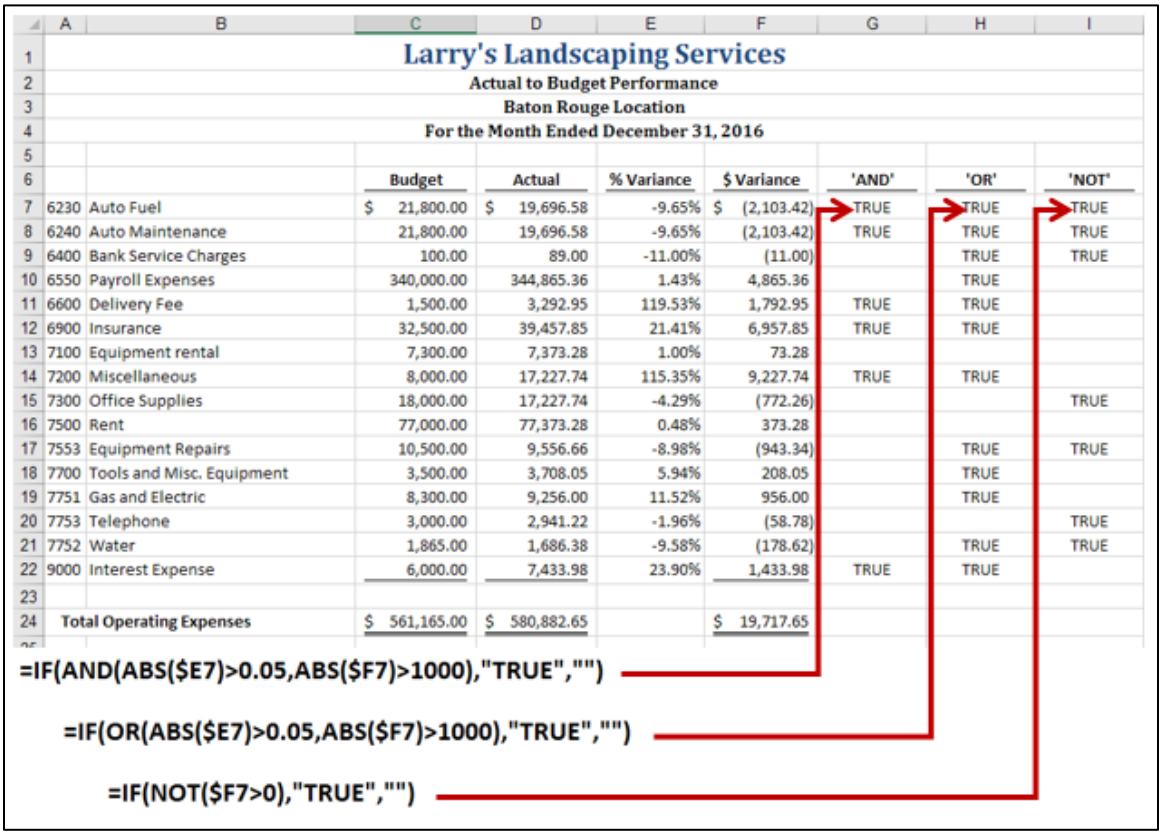
IF(logical test, value if true, value if false)

The value returned by the function depends on the results of the logical test. In the example in **Figure 7**, a formula based on the IF function is used to calculate a budget variance percentage, but only if the budget amount is not equal to zero, thereby avoiding the **#DIV/0** error. If the budget amount is equal to zero, the formula returns a null string, which displays a blank cell. To return a null string, use two quotation marks entered side-by-side as shown in the sample formula.

```
=IF($C7<>0,($D7-$C7)/$C7,"")
```

Figure 7 - Formula to Return Null String If Budget Equals Zero

The Boolean functions – **AND**, **OR**, and **NOT** – can be used in combination with other functions to produce compound conditional formulas. For example, to create a formula that requires multiple conditions to be met in order to return a specified result, use AND in conjunction with the IF function as shown in **Figure 8**.



	A	B	C	D	E	F	G	H	I
1		Larry's Landscaping Services							
2		Actual to Budget Performance							
3		Baton Rouge Location							
4		For the Month Ended December 31, 2016							
5									
6			Budget	Actual	% Variance	$ Variance	'AND'	'OR'	'NOT'
7	6230	Auto Fuel	$ 21,800.00	$ 19,696.58	-9.65%	$ (2,103.42)	TRUE	TRUE	TRUE
8	6240	Auto Maintenance	21,800.00	19,696.58	-9.65%	(2,103.42)	TRUE	TRUE	TRUE
9	6400	Bank Service Charges	100.00	89.00	-11.00%	(11.00)		TRUE	TRUE
10	6550	Payroll Expenses	340,000.00	344,865.36	1.43%	4,865.36		TRUE	
11	6600	Delivery Fee	1,500.00	3,292.95	119.53%	1,792.95	TRUE	TRUE	
12	6900	Insurance	32,500.00	39,457.85	21.41%	6,957.85	TRUE	TRUE	
13	7100	Equipment rental	7,300.00	7,373.28	1.00%	73.28			
14	7200	Miscellaneous	8,000.00	17,227.74	115.35%	9,227.74	TRUE	TRUE	
15	7300	Office Supplies	18,000.00	17,227.74	-4.29%	(772.26)			TRUE
16	7500	Rent	77,000.00	77,373.28	0.48%	373.28			
17	7553	Equipment Repairs	10,500.00	9,556.66	-8.98%	(943.34)		TRUE	TRUE
18	7700	Tools and Misc. Equipment	3,500.00	3,708.05	5.94%	208.05		TRUE	
19	7751	Gas and Electric	8,300.00	9,256.00	11.52%	956.00		TRUE	
20	7753	Telephone	3,000.00	2,941.22	-1.96%	(58.78)			TRUE
21	7752	Water	1,865.00	1,686.38	-9.58%	(178.62)		TRUE	TRUE
22	9000	Interest Expense	6,000.00	7,433.98	23.90%	1,433.98	TRUE	TRUE	
23									
24		Total Operating Expenses	$ 561,165.00	$ 580,882.65		$ 19,717.65			

Figure 8 - Using Boolean Functions to Build Compound Conditional Formulas

Conditional IF Functions

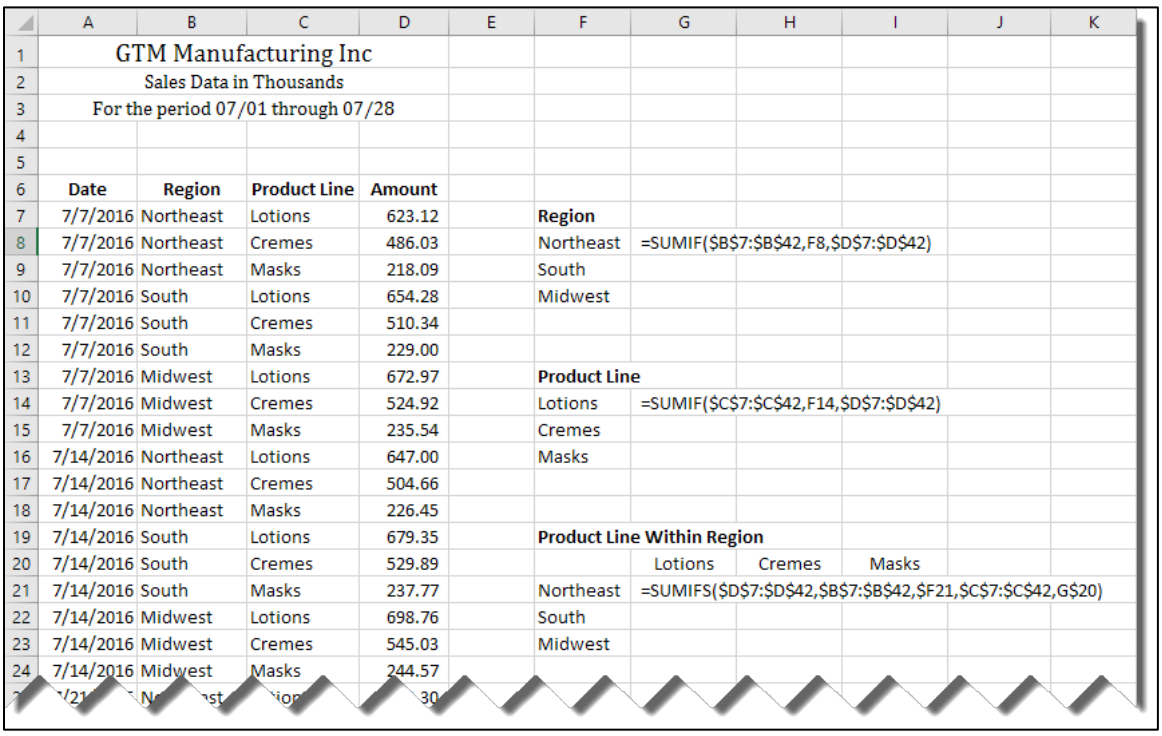
Excel includes built-in functions to do conditional sums, counts, and averages – **SUMIF**, **SUMIFS**, **COUNTIF**, **COUNTIFS**, **AVERAGEIF**, and **AVERAGEIFS**. The singular versions of these functions (SUMIF, COUNTIF, and AVERAGEIF) only support a single conditional test and are similar in application, operation, and syntax. The plural versions of these functions (SUMIFS, COUNTIFS, and AVERAGEIFS) do conditional sums, counts, and averages with up to 127 conditions. Their syntax is slightly different. For example, compare the syntax of SUMIF with SUMIFS.

SUMIF(range, criteria, sum range)
SUMIFS(sum range, criteria range1, criteria1, criteria range2,criteria2…)

Note that the *sum range* is the last criterion in the SUMIF function, but the first criterion in the SUMIFS function. SUMIFS can support criterion references for up to 127 criteria. Similar to SUMIF, criteria can be entered as numbers, cell references, or text surrounded by quotation marks and can contain wildcard characters. However, criteria ranges *must* be the same shape and size as the sum range.

GTM Manufacturing Company compiles a list of weekly sales by region and product. The CFO needs to produce a report each month that sums the sales by product, by region, and by product within region. In the past, she has sorted and summed the data multiple times in order to create and paste the desired totals to her report sheet. In Excel, she can create the report with three simple formulas – one SUMIF to total sales by region, one SUMIF to total sales by product, and one SUMIFS to sum sales by product within region. The SUMIF formulas will use *absolute* addresses to define the sum and criteria ranges so that the formula can be copied down without error.

The SUMIFS formula will use *absolute* addresses to define the sum and criteria ranges, but it will use *mixed* addresses for the criteria so that the formula can be copied down *and* across. **Figure 9** displays the worksheet used to produce the required totals.



	A	B	C	D	E	F	G	H	I	J	K
1	GTM Manufacturing Inc										
2	Sales Data in Thousands										
3	For the period 07/01 through 07/28										
4											
5											
6	**Date**	**Region**	**Product Line**	**Amount**							
7	7/7/2016	Northeast	Lotions	623.12		**Region**					
8	7/7/2016	Northeast	Cremes	486.03		Northeast	=SUMIF($B$7:$B$42,F8,$D$7:$D$42)				
9	7/7/2016	Northeast	Masks	218.09		South					
10	7/7/2016	South	Lotions	654.28		Midwest					
11	7/7/2016	South	Cremes	510.34							
12	7/7/2016	South	Masks	229.00							
13	7/7/2016	Midwest	Lotions	672.97		**Product Line**					
14	7/7/2016	Midwest	Cremes	524.92		Lotions	=SUMIF($C$7:$C$42,F14,$D$7:$D$42)				
15	7/7/2016	Midwest	Masks	235.54		Cremes					
16	7/14/2016	Northeast	Lotions	647.00		Masks					
17	7/14/2016	Northeast	Cremes	504.66							
18	7/14/2016	Northeast	Masks	226.45							
19	7/14/2016	South	Lotions	679.35		**Product Line Within Region**					
20	7/14/2016	South	Cremes	529.89			Lotions	Cremes	Masks		
21	7/14/2016	South	Masks	237.77		Northeast	=SUMIFS($D$7:$D$42,$B$7:$B$42,$F21,$C$7:$C$42,G$20)				
22	7/14/2016	Midwest	Lotions	698.76		South					
23	7/14/2016	Midwest	Cremes	545.03		Midwest					
24	7/14/2016	Midwest	Masks	244.57							

Figure 9 - Using SUMIF and SUMIFS to Produce a Sales Report

The formula to produce sales totals by product within region is shown in **Figure 20**. Note the use of mixed addresses in the formula for the criteria. In this formula, the criterion for ***Region*** is $F21, which is absolute as to column and relative as to row. As the formula is copied across columns, the formula will always look to column F for the ***Region*** criterion relative to the row in which the formula resides. The criterion for ***Product Line*** is G$20, which is relative as to column and absolute as to row. As the formula is copied down across rows, the formula will always look to row 20 for the ***Product Line*** criterion relative to the column in which the formula resides. Stipulating *mixed* addresses for the criteria is the magic that allows the formula to be copied down and across.

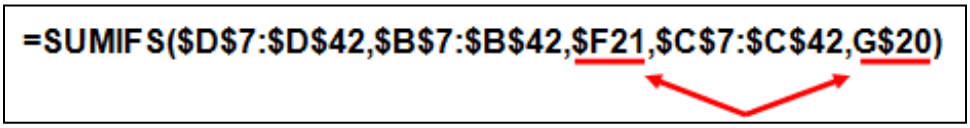


Figure 10 - Using Mixed Addresses in Formulas

Summarizing Data with SUBTOTAL

Many accounting workbooks have multiple levels of subtotals that are ultimately summarized into a grand total. The **SUBTOTAL** function makes this process easier and less prone to error, especially when using big datasets with many subtotals. Nested subtotals (subtotals within subtotals) are ignored to avoid double counting. Grand totals are always calculated on the detail data, not on any included intermediate (nested) subtotals. The SUBTOTAL function can summarize *columns* of data using any of eleven methods identified in **Table 1**.

	Function Number	
Function	**Includes Hidden Values**	**Ignores Hidden Values**
AVERAGE	1	101
COUNT	2	102
COUNTA	3	103
MAX	4	104

MIN	5	105
PRODUCT	6	106
STDEV	7	107
STDEVP	8	108
SUM	9	109
VAR	10	110
VARP	11	111

Table 1 - SUBTOTAL Functions in Excel

Note that there are two columns of functions in the table. The first indicates that values on hidden rows in the subtotal are included in the calculation. The second column ignores values on hidden rows.

Data Filters alter the rules of the SUBTOTAL function. Summarizations calculated on filtered data always exclude values on hidden rows regardless of the function used. In other words, functions 1 and 101 do not include values on hidden rows in the calculation of subtotals. Furthermore, operation of the **AutoSum** button accessible on the **Home** tab of the Ribbon is altered when a *filter is active* so that SUBTOTAL functions are entered rather than simple summarization functions, such as SUM, AVERAGE, or COUNT, etc.

The example in **Figure 11** uses the SUBTOTAL function to produce subtotals on rows 11 and 19 and a grand total on row 21. The formulas in the left column of amounts uses function **9** in the SUBTOTAL function. These formulas include hidden values in the subtotals. The formulas in the right column of amounts use function **109** in the SUBTOTAL function. These formulas do not include hidden values in the subtotals. If function 109 is used, it is a simple task to create a report that excludes miscellaneous expenses. Just hide the two miscellaneous expense rows, and the calculations of the subtotals and grand total will immediately reflect their absence.

	A	B	C	D	E
1	**Departmental Expenses**				
2					
3			**Function Used**		
4			**9**	**109**	
5	**Tax Department**				
6	8000	Meals and Entertainment	$ 865.35	$ 865.35	
7	8020	Airfares	7,864.36	7,864.36	
8	8030	Hotels	2,899.36	2,899.36	
9	8040	Local Transportation	648.65	648.65	
10	8050	Miscellaneous	483.21	483.21	
11	Total for the Tax Department		$12,760.93	$12,760.93	
12			**=SUBTOTAL(9,C5:C10)**	**=SUBTOTAL(109,D5:D10)**	
13	**Audit Department**				
14	8100	Meals and Entertainment	$ 579.78	$ 579.78	
15	8120	Airfares	5,269.12	5,269.12	
16	8130	Hotels	1,942.57	1,942.57	
17	8140	Local Transportation	434.60	434.60	
18	8150	Miscellaneous	323.75	323.75	
19	Total for the Audit Department		$ 8,549.82	$ 8,549.82	
20			**=SUBTOTAL(9,C13:C18)**	**=SUBTOTAL(109,D13:D18)**	
21		Total for All Departments	$21,310.75	$21,310.75	
22			**=SUBTOTAL(9,C5:C19)**	**=SUBTOTAL(109,D5:D19)**	

Figure 11 - Using SUBTOTAL to Automate the Totaling Process