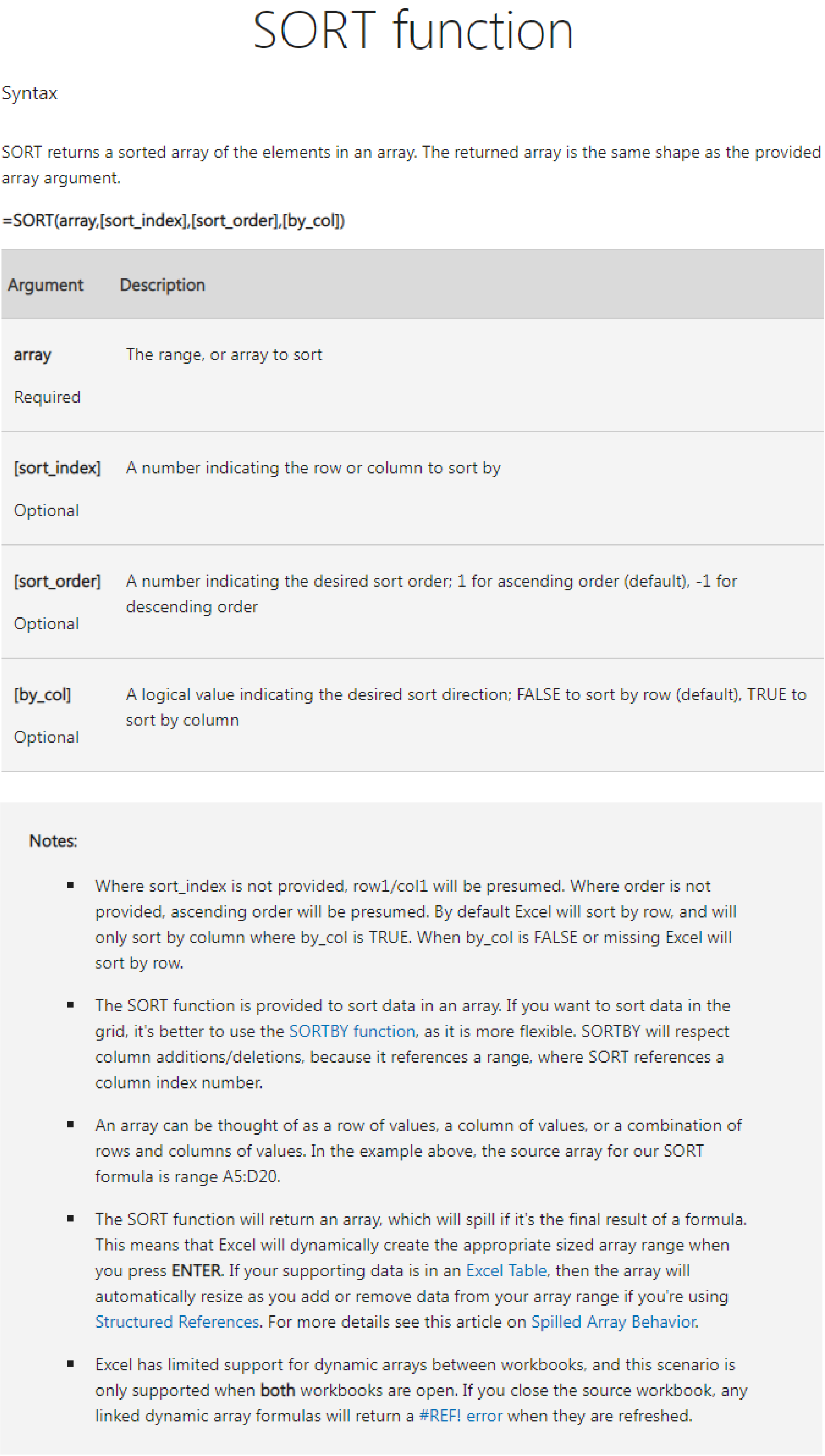
1. SORT 2
2. SORTBY 4
3. FILTER 7
4. UNIQUE 10



SORT function

Syntax

SORT returns a sorted array of the elements in an array. The returned array is the same shape as the provided array argument.

=SORT(array,[sort_index],[sort_order],[by_col])

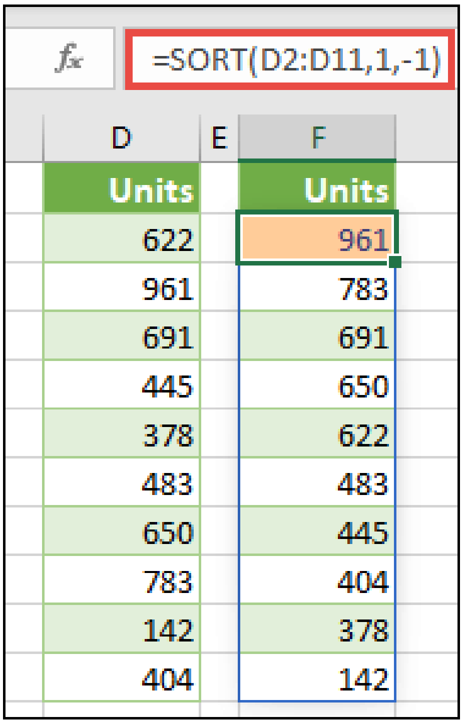
Argument	Description
array Required	The range, or array to sort
[sort_index] Optional	A number indicating the row or column to sort by
[sort_order] Optional	A number indicating the desired sort order; 1 for ascending order (default), -1 for descending order
[by_col] Optional	A logical value indicating the desired sort direction; FALSE to sort by row (default), TRUE to sort by column

Notes:

- Where sort_index is not provided, row1/col1 will be presumed. Where order is not provided, ascending order will be presumed. By default Excel will sort by row, and will only sort by column where by_col is TRUE. When by_col is FALSE or missing Excel will sort by row.
- The SORT function is provided to sort data in an array. If you want to sort data in the grid, it's better to use the SORTBY function, as it is more flexible. SORTBY will respect column additions/deletions, because it references a range, where SORT references a column index number.
- An array can be thought of as a row of values, a column of values, or a combination of rows and columns of values. In the example above, the source array for our SORT formula is range A5:D20.
- The SORT function will return an array, which will spill if it's the final result of a formula. This means that Excel will dynamically create the appropriate sized array range when you press **ENTER**. If your supporting data is in an Excel Table, then the array will automatically resize as you add or remove data from your array range if you're using Structured References. For more details see this article on Spilled Array Behavior.
- Excel has limited support for dynamic arrays between workbooks, and this scenario is only supported when **both** workbooks are open. If you close the source workbook, any linked dynamic array formulas will return a #REF! error when they are refreshed.

Examples

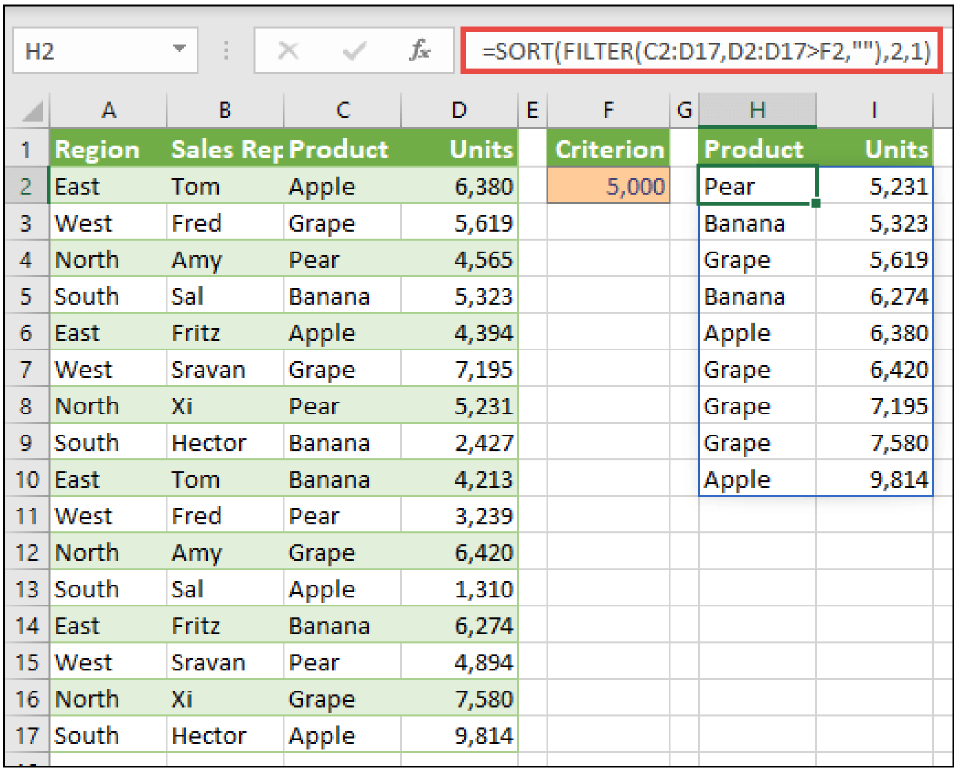
Sort a range of values in descending order.



=SORT(D2:D11,1,-1)

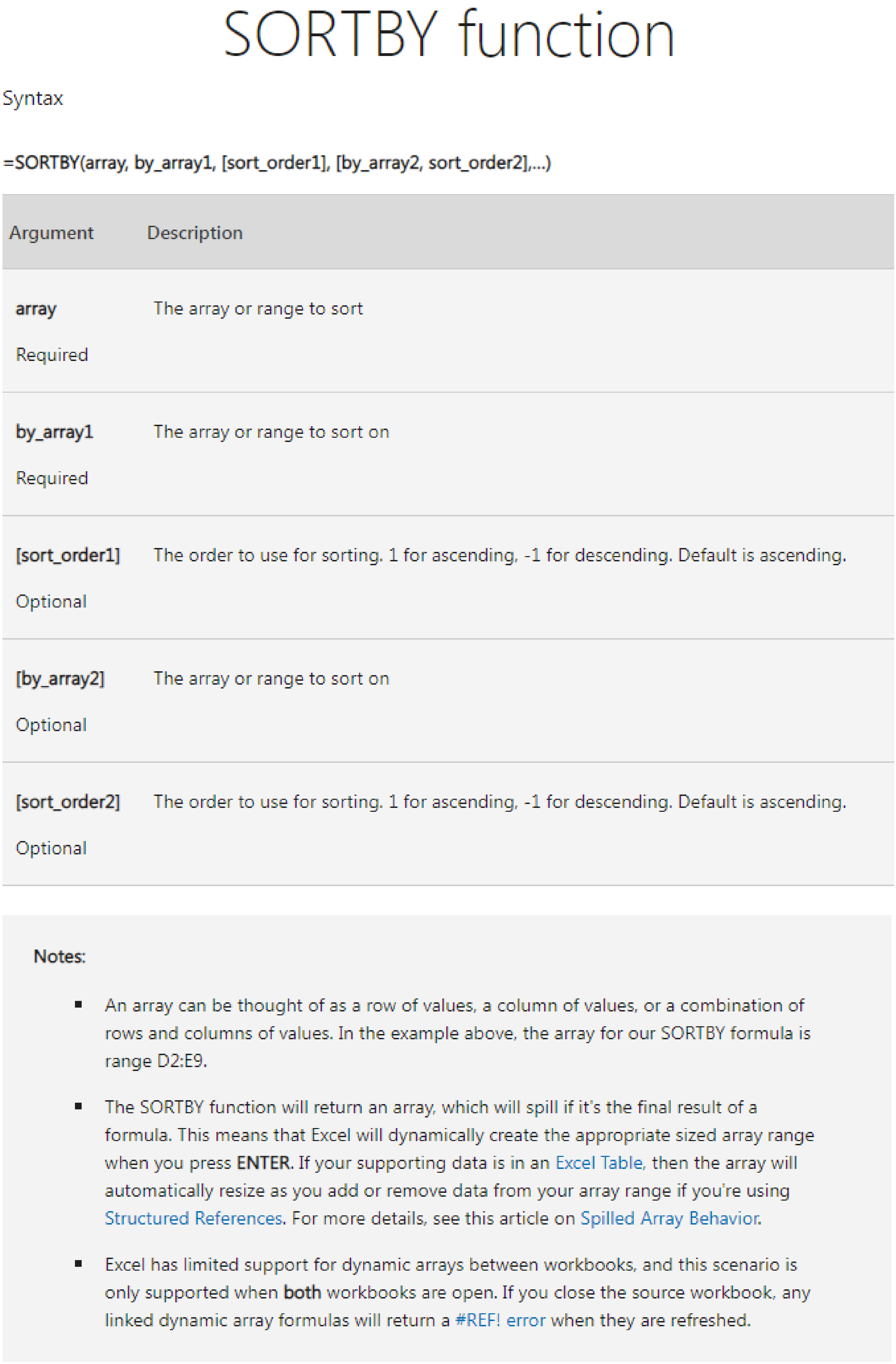
D	E	F
Units		Units
622		961
961		783
691		691
445		650
378		622
483		483
650		445
783		404
142		378
404		142

Use SORT and FILTER together to sort a range in ascending order, and limit it to values over 5,000.



H2 =SORT(FILTER(C2:D17,D2:D17>F2,""),2,1)

	A	B	C	D	E	F	G	H	I
1	Region	Sales Rep	Product	Units		Criterion		Product	Units
2	East	Tom	Apple	6,380		5,000		Pear	5,231
3	West	Fred	Grape	5,619				Banana	5,323
4	North	Amy	Pear	4,565				Grape	5,619
5	South	Sal	Banana	5,323				Banana	6,274
6	East	Fritz	Apple	4,394				Apple	6,380
7	West	Sravan	Grape	7,195				Grape	6,420
8	North	Xi	Pear	5,231				Grape	7,195
9	South	Hector	Banana	2,427				Grape	7,580
10	East	Tom	Banana	4,213				Apple	9,814
11	West	Fred	Pear	3,239					
12	North	Amy	Grape	6,420					
13	South	Sal	Apple	1,310					
14	East	Fritz	Banana	6,274					
15	West	Sravan	Pear	4,894					
16	North	Xi	Grape	7,580					
17	South	Hector	Apple	9,814					



SORTBY function

Syntax

=SORTBY(array, by_array1, [sort_order1], [by_array2, sort_order2],…)

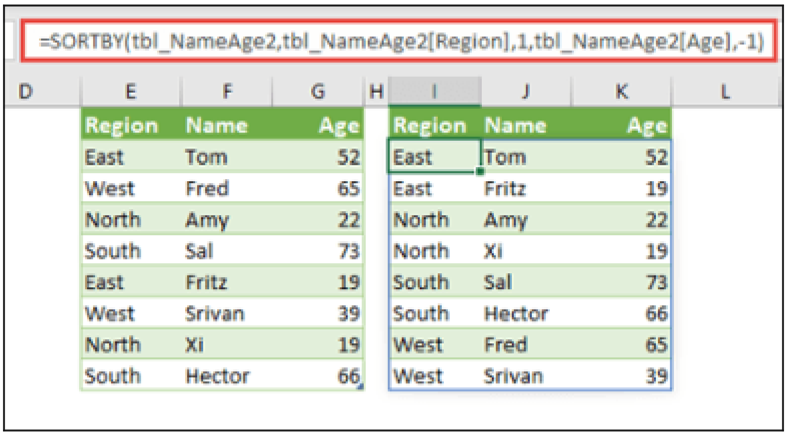
Argument	Description
array Required	The array or range to sort
by_array1 Required	The array or range to sort on
[sort_order1] Optional	The order to use for sorting. 1 for ascending, -1 for descending. Default is ascending.
[by_array2] Optional	The array or range to sort on
[sort_order2] Optional	The order to use for sorting. 1 for ascending, -1 for descending. Default is ascending.

Notes:

- An array can be thought of as a row of values, a column of values, or a combination of rows and columns of values. In the example above, the array for our SORTBY formula is range D2:E9.
- The SORTBY function will return an array, which will spill if it's the final result of a formula. This means that Excel will dynamically create the appropriate sized array range when you press **ENTER**. If your supporting data is in an Excel Table, then the array will automatically resize as you add or remove data from your array range if you're using Structured References. For more details, see this article on Spilled Array Behavior.
- Excel has limited support for dynamic arrays between workbooks, and this scenario is only supported when **both** workbooks are open. If you close the source workbook, any linked dynamic array formulas will return a #REF! error when they are refreshed.

Examples

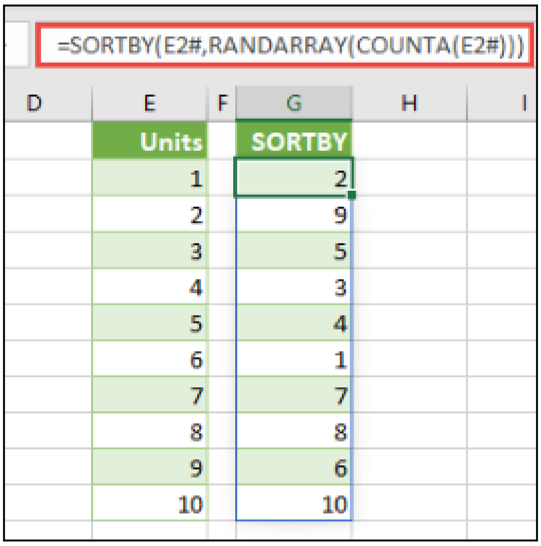
Sort a table by Region in ascending order, then by each person's age, in descending order.



=SORTBY(tbl_NameAge2,tbl_NameAge2[Region],1,tbl_NameAge2[Age],-1)

Region	Name	Age	Region	Name	Age
East	Tom	52	East	Tom	52
West	Fred	65	East	Fritz	19
North	Amy	22	North	Amy	22
South	Sal	73	North	Xi	19
East	Fritz	19	South	Sal	73
West	Srivan	39	South	Hector	66
North	Xi	19	West	Fred	65
South	Hector	66	West	Srivan	39

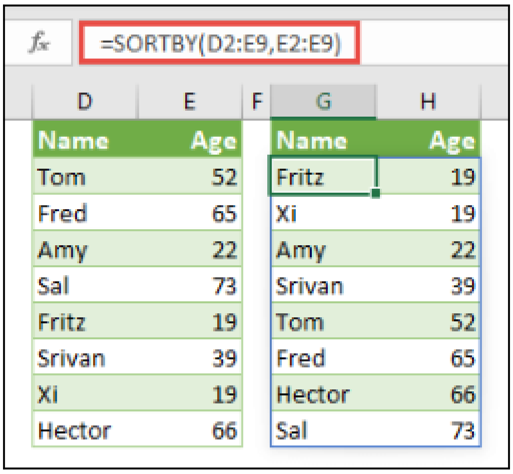
Use SORTBY with RANDARRAY, and COUNTA to randomize a list of values. In this case, E2# references the dynamic array range beginning in cell E2, as that was populated by using =SEQUENCE(10). The # sign is called the spilled range operator.



=SORTBY(E2#,RANDARRAY(COUNTA(E2#)))

Units	SORTBY
1	2
2	9
3	5
4	3
5	4
6	1
7	7
8	8
9	6
10	10

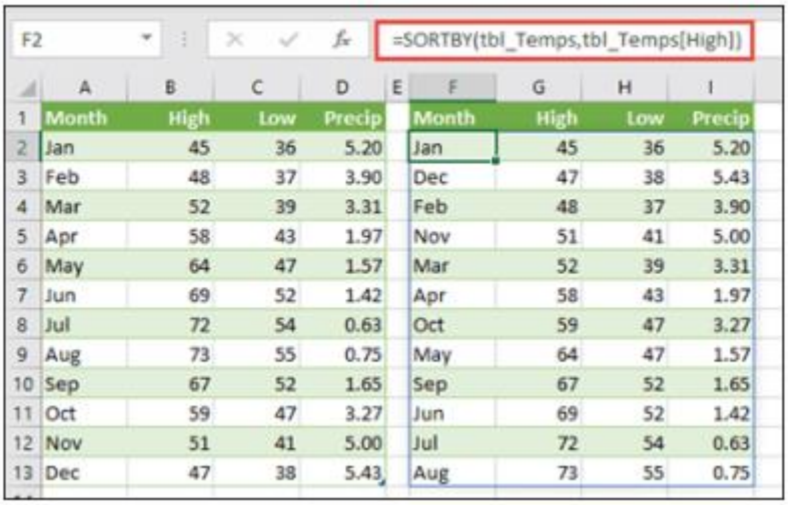
In this example, we're sorting a list of people's names by their age, in ascending order.



=SORTBY(D2:E9,E2:E9)

Name	Age	Name	Age
Tom	52	Fritz	19
Fred	65	Xi	19
Amy	22	Amy	22
Sal	73	Srivan	39
Fritz	19	Tom	52
Srivan	39	Fred	65
Xi	19	Hector	66
Hector	66	Sal	73

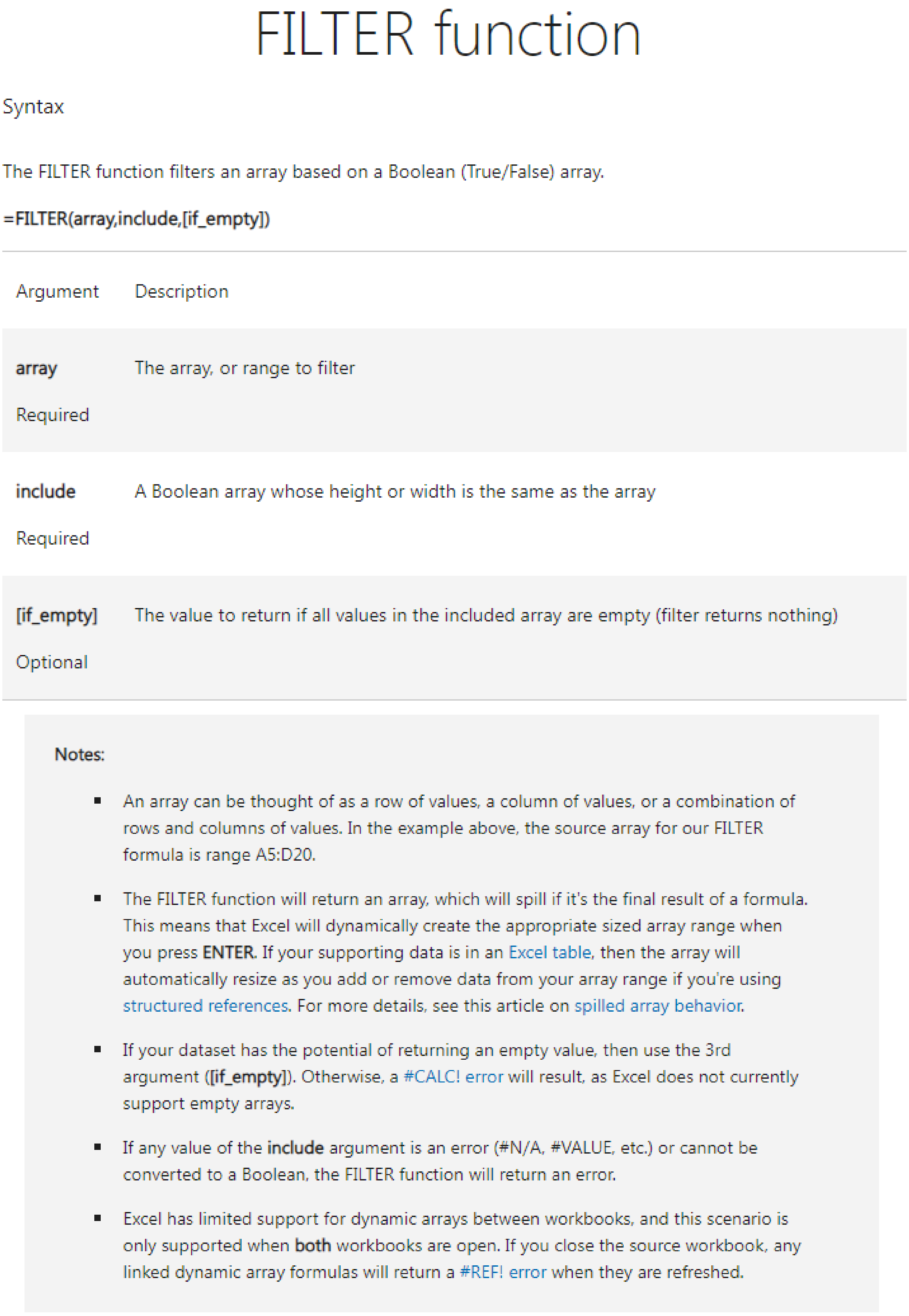
Use SORTBY to sort a table of temperature and rainfall values by high temperature.



F2

=SORTBY(tbl_Temps,tbl_Temps[High])

	A	B	C	D	E	F	G	H	I
1	Month	High	Low	Precip		Month	High	Low	Precip
2	Jan	45	36	5.20		Jan	45	36	5.20
3	Feb	48	37	3.90		Dec	47	38	5.43
4	Mar	52	39	3.31		Feb	48	37	3.90
5	Apr	58	43	1.97		Nov	51	41	5.00
6	May	64	47	1.57		Mar	52	39	3.31
7	Jun	69	52	1.42		Apr	58	43	1.97
8	Jul	72	54	0.63		Oct	59	47	3.27
9	Aug	73	55	0.75		May	64	47	1.57
10	Sep	67	52	1.65		Sep	67	52	1.65
11	Oct	59	47	3.27		Jun	69	52	1.42
12	Nov	51	41	5.00		Jul	72	54	0.63
13	Dec	47	38	5.43		Aug	73	55	0.75



FILTER function

Syntax

The FILTER function filters an array based on a Boolean (True/False) array.

=FILTER(array,include,[if_empty])

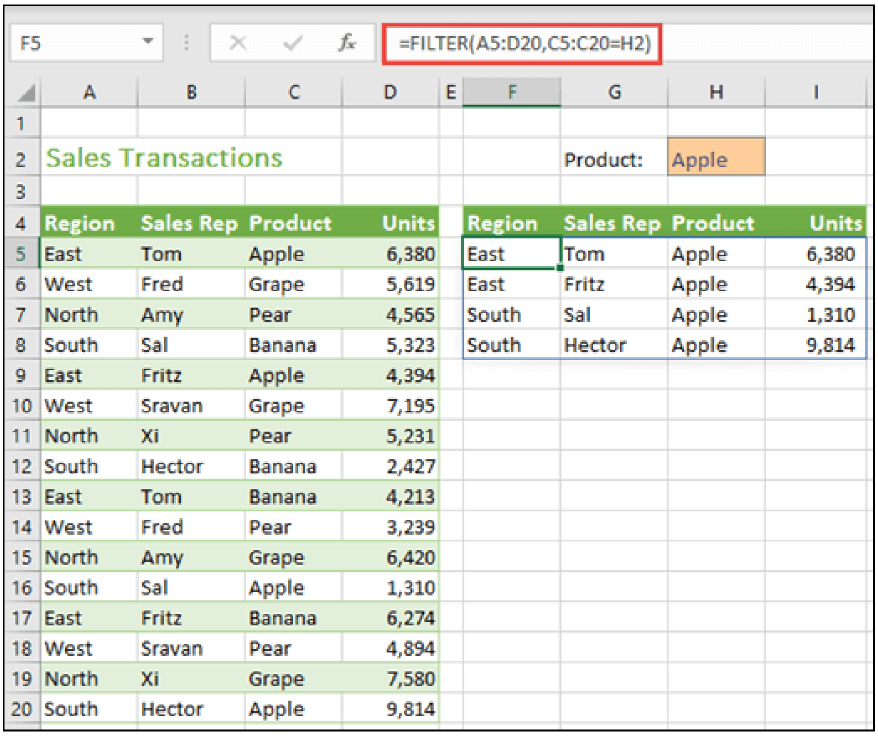
Argument	Description
array Required	The array, or range to filter
include Required	A Boolean array whose height or width is the same as the array
[if_empty] Optional	The value to return if all values in the included array are empty (filter returns nothing)

Notes:

- An array can be thought of as a row of values, a column of values, or a combination of rows and columns of values. In the example above, the source array for our FILTER formula is range A5:D20.
- The FILTER function will return an array, which will spill if it's the final result of a formula. This means that Excel will dynamically create the appropriate sized array range when you press **ENTER**. If your supporting data is in an Excel table, then the array will automatically resize as you add or remove data from your array range if you're using structured references. For more details, see this article on spilled array behavior.
- If your dataset has the potential of returning an empty value, then use the 3rd argument (**[if_empty]**). Otherwise, a #CALC! error will result, as Excel does not currently support empty arrays.
- If any value of the **include** argument is an error (#N/A, #VALUE, etc.) or cannot be converted to a Boolean, the FILTER function will return an error.
- Excel has limited support for dynamic arrays between workbooks, and this scenario is only supported when **both** workbooks are open. If you close the source workbook, any linked dynamic array formulas will return a #REF! error when they are refreshed.

Examples

In the following example we used the formula **=FILTER(A5:D20,C5:C20=H2,"")** to return all records for Apple, as selected in cell H2, and if there are no apples, return an empty string ("").



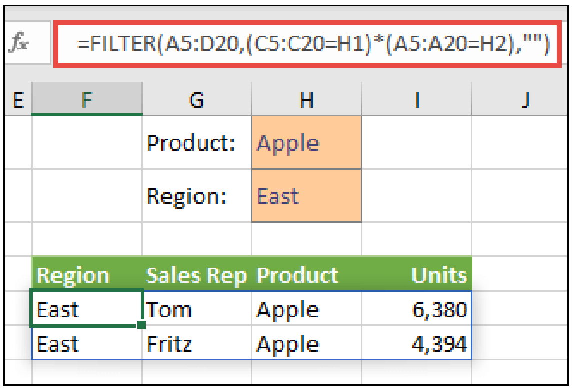
F5

=FILTER(A5:D20,C5:C20=H2)

	A	B	C	D	E	F	G	H	I
1									
2	Sales Transactions						Product:	Apple	
3									
4	Region	Sales Rep	Product	Units		Region	Sales Rep	Product	Units
5	East	Tom	Apple	6,380		East	Tom	Apple	6,380
6	West	Fred	Grape	5,619		East	Fritz	Apple	4,394
7	North	Amy	Pear	4,565		South	Sal	Apple	1,310
8	South	Sal	Banana	5,323		South	Hector	Apple	9,814
9	East	Fritz	Apple	4,394					
10	West	Sravan	Grape	7,195					
11	North	Xi	Pear	5,231					
12	South	Hector	Banana	2,427					
13	East	Tom	Banana	4,213					
14	West	Fred	Pear	3,239					
15	North	Amy	Grape	6,420					
16	South	Sal	Apple	1,310					
17	East	Fritz	Banana	6,274					
18	West	Sravan	Pear	4,894					
19	North	Xi	Grape	7,580					
20	South	Hector	Apple	9,814					

FILTER used to return multiple criteria

In this case, we're using the multiplication operator (*) to return all values in our array range (A5:D20) that have Apples **AND** are in the East region: **=FILTER(A5:D20,(C5:C20=H1)*(A5:A20=H2),"")**.

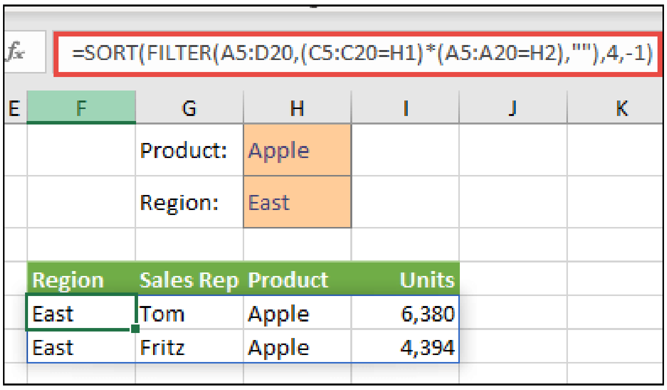


=FILTER(A5:D20,(C5:C20=H1)*(A5:A20=H2),"")

E	F	G	H	I	J
		Product:	Apple		
		Region:	East		
	Region	Sales Rep	Product	Units	
	East	Tom	Apple	6,380	
	East	Fritz	Apple	4,394	

FILTER used to return multiple criteria and sort

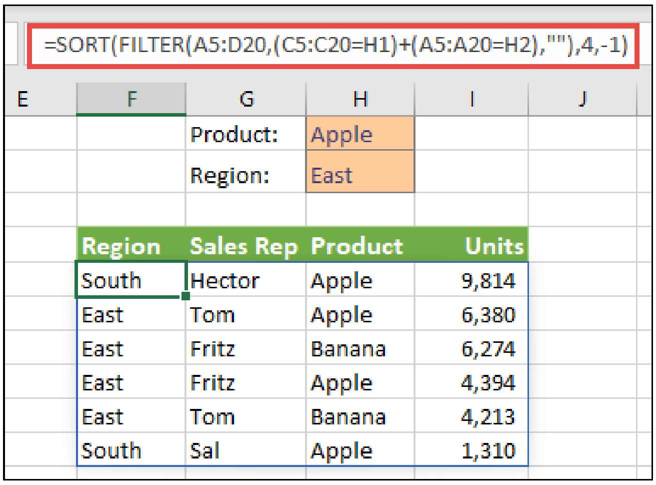
In this case, we're using the previous FILTER function with the SORT function to return all values in our array range (A5:D20) that have Apples **AND** are in the East region, and then sort Units in descending order: **=SORT(FILTER(A5:D20,(C5:C20=H1)*(A5:A20=H2),""),4,-1)**



=SORT(FILTER(A5:D20,(C5:C20=H1)*(A5:A20=H2),""),4,-1)

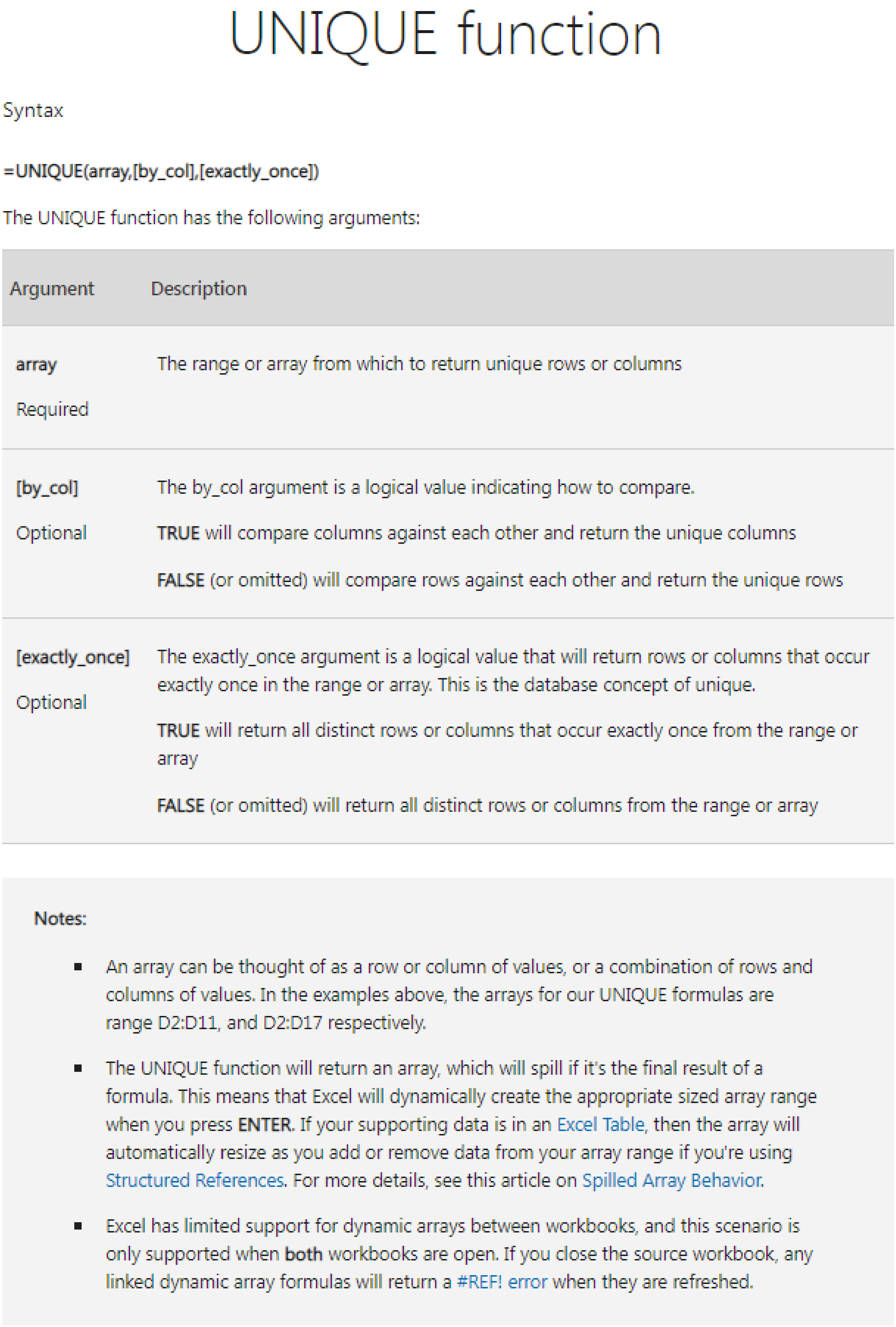
E	F	G	H	I	J	K
		Product:	Apple			
		Region:	East			
	Region	Sales Rep	Product	Units		
	East	Tom	Apple	6,380		
	East	Fritz	Apple	4,394		

In this case, we're using the FILTER function with the addition operator (+) to return all values in our array range (A5:D20) that have Apples **OR** are in the East region, and then sort Units in descending order: **=SORT(FILTER(A5:D20,(C5:C20=H1)+(A5:A20=H2),""),4,-1)**.



=SORT(FILTER(A5:D20,(C5:C20=H1)+(A5:A20=H2),""),4,-1)

E	F	G	H	I	J
		Product:	Apple		
		Region:	East		
	Region	Sales Rep	Product	Units	
	South	Hector	Apple	9,814	
	East	Tom	Apple	6,380	
	East	Fritz	Banana	6,274	
	East	Fritz	Apple	4,394	
	East	Tom	Banana	4,213	
	South	Sal	Apple	1,310	



UNIQUE function

Syntax

=UNIQUE(array,[by_col],[exactly_once])

The UNIQUE function has the following arguments:

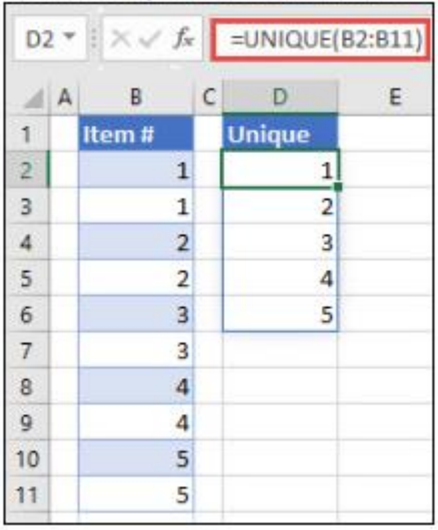
Argument	Description
array Required	The range or array from which to return unique rows or columns
[by_col] Optional	The by_col argument is a logical value indicating how to compare. **TRUE** will compare columns against each other and return the unique columns **FALSE** (or omitted) will compare rows against each other and return the unique rows
[exactly_once] Optional	The exactly_once argument is a logical value that will return rows or columns that occur exactly once in the range or array. This is the database concept of unique. **TRUE** will return all distinct rows or columns that occur exactly once from the range or array **FALSE** (or omitted) will return all distinct rows or columns from the range or array

Notes:

- An array can be thought of as a row or column of values, or a combination of rows and columns of values. In the examples above, the arrays for our UNIQUE formulas are range D2:D11, and D2:D17 respectively.
- The UNIQUE function will return an array, which will spill if it's the final result of a formula. This means that Excel will dynamically create the appropriate sized array range when you press **ENTER**. If your supporting data is in an Excel Table, then the array will automatically resize as you add or remove data from your array range if you're using Structured References. For more details, see this article on Spilled Array Behavior.
- Excel has limited support for dynamic arrays between workbooks, and this scenario is only supported when **both** workbooks are open. If you close the source workbook, any linked dynamic array formulas will return a #REF! error when they are refreshed.

Examples

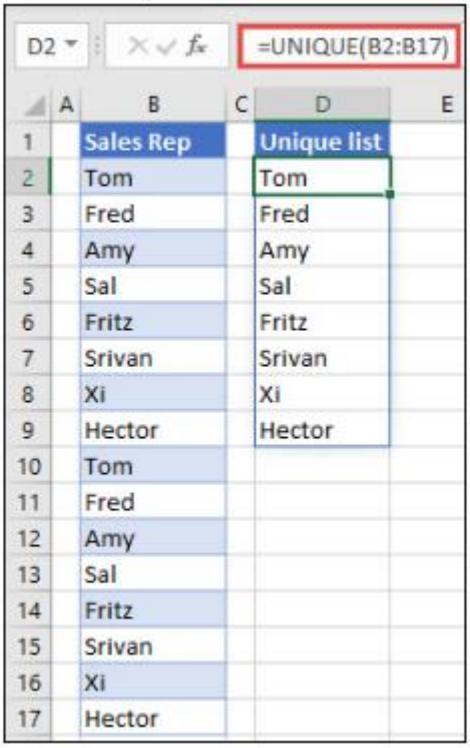
Return unique values from a list of values



=UNIQUE(B2:B11)

	A	B	C	D	E
1		Item #		Unique	
2		1		1	
3		1		2	
4		2		3	
5		2		4	
6		3		5	
7		3			
8		4			
9		4			
10		5			
11		5			

Return unique names from a list of names

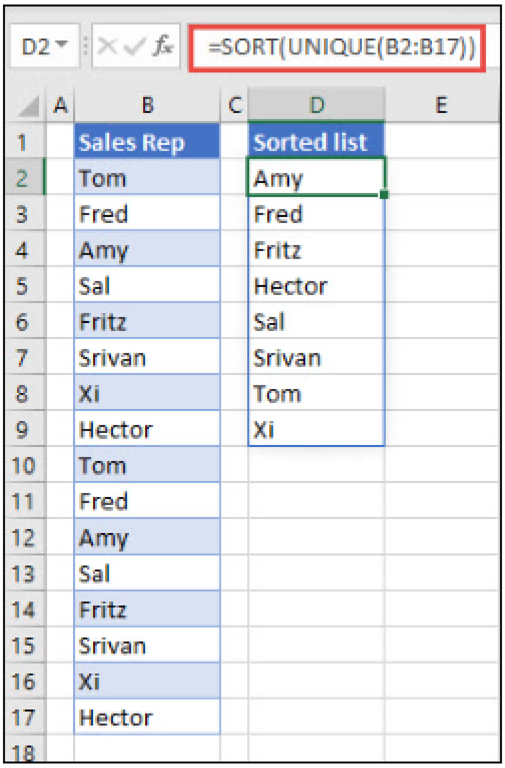


=UNIQUE(B2:B17)

	A	B	C	D	E
1		Sales Rep		Unique list	
2		Tom		Tom	
3		Fred		Fred	
4		Amy		Amy	
5		Sal		Sal	
6		Fritz		Fritz	
7		Srivan		Srivan	
8		Xi		Xi	
9		Hector		Hector	
10		Tom			
11		Fred			
12		Amy			
13		Sal			
14		Fritz			
15		Srivan			
16		Xi			
17		Hector			

Example 1

This example uses SORT and UNIQUE together to return a unique list of names in ascending order.

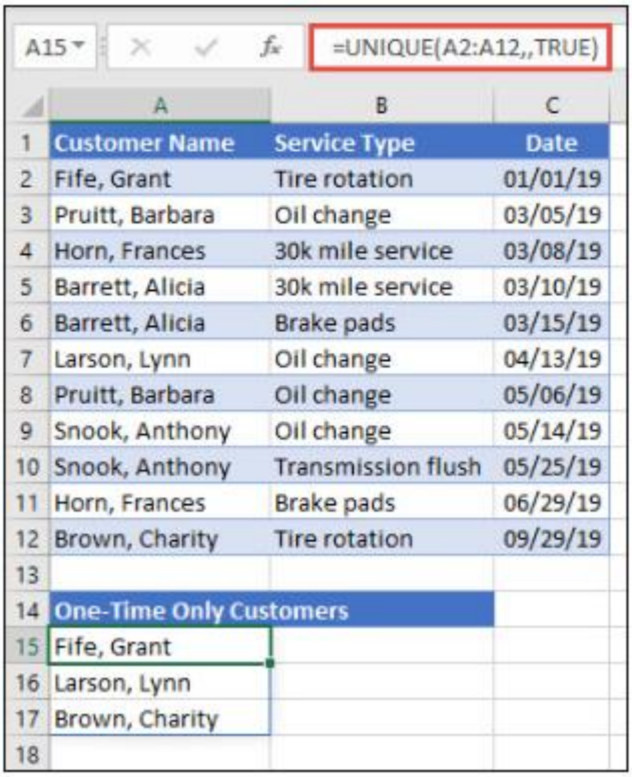


=SORT(UNIQUE(B2:B17))

	A	B	C	D	E
1		Sales Rep		Sorted list	
2		Tom		Amy	
3		Fred		Fred	
4		Amy		Fritz	
5		Sal		Hector	
6		Fritz		Sal	
7		Srivan		Srivan	
8		Xi		Tom	
9		Hector		Xi	
10		Tom			
11		Fred			
12		Amy			
13		Sal			
14		Fritz			
15		Srivan			
16		Xi			
17		Hector			
18					

Example 2

This example has the exactly_once argument set to TRUE, and the function returns only those customers who have had service one time. This can be useful if you want to identify people who have not returned for additional service, so you can contact them.

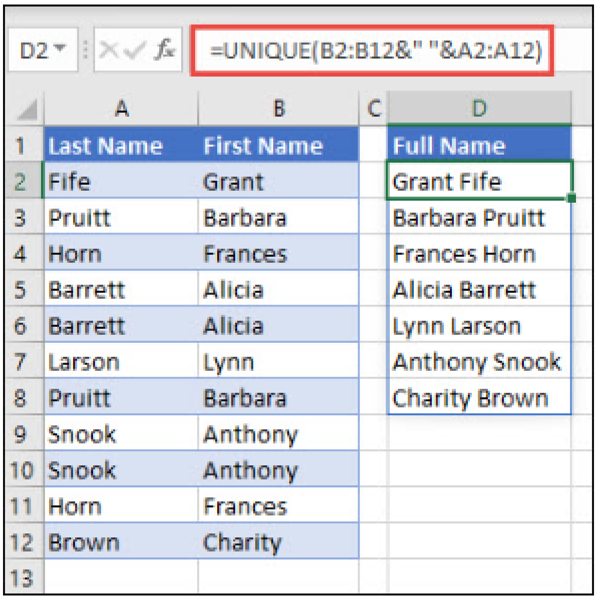


=UNIQUE(A2:A12,,TRUE)

	A	B	C
1	Customer Name	Service Type	Date
2	Fife, Grant	Tire rotation	01/01/19
3	Pruitt, Barbara	Oil change	03/05/19
4	Horn, Frances	30k mile service	03/08/19
5	Barrett, Alicia	30k mile service	03/10/19
6	Barrett, Alicia	Brake pads	03/15/19
7	Larson, Lynn	Oil change	04/13/19
8	Pruitt, Barbara	Oil change	05/06/19
9	Snook, Anthony	Oil change	05/14/19
10	Snook, Anthony	Transmission flush	05/25/19
11	Horn, Frances	Brake pads	06/29/19
12	Brown, Charity	Tire rotation	09/29/19
13			
14	One-Time Only Customers		
15	Fife, Grant		
16	Larson, Lynn		
17	Brown, Charity		
18			

Example 3

This example uses the ampersand (&) to concatenate last name and first name into a full name. Note that the formula references the entire range of names in A2:A12 and B2:B12. This allows Excel to return an array of all names.

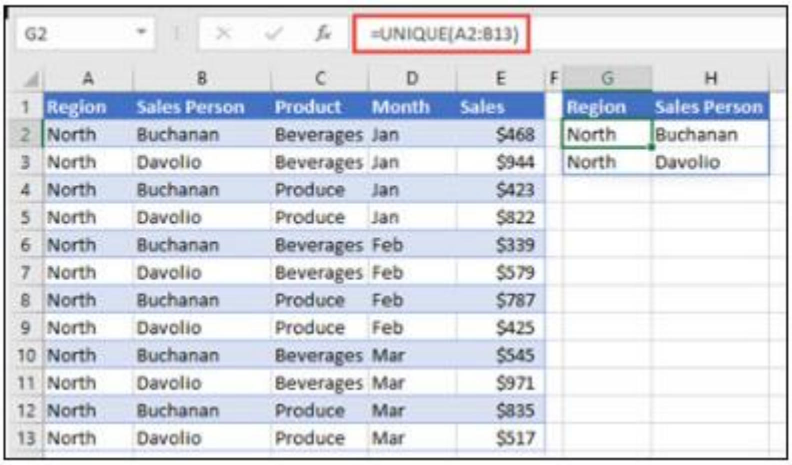


D2 =UNIQUE(B2:B12&" "&A2:A12)

	A	B	C	D
1	Last Name	First Name		Full Name
2	Fife	Grant		Grant Fife
3	Pruitt	Barbara		Barbara Pruitt
4	Horn	Frances		Frances Horn
5	Barrett	Alicia		Alicia Barrett
6	Barrett	Alicia		Lynn Larson
7	Larson	Lynn		Anthony Snook
8	Pruitt	Barbara		Charity Brown
9	Snook	Anthony		
10	Snook	Anthony		
11	Horn	Frances		
12	Brown	Charity		
13				

Example 4

This example compares two columns and returns only the unique values between them.

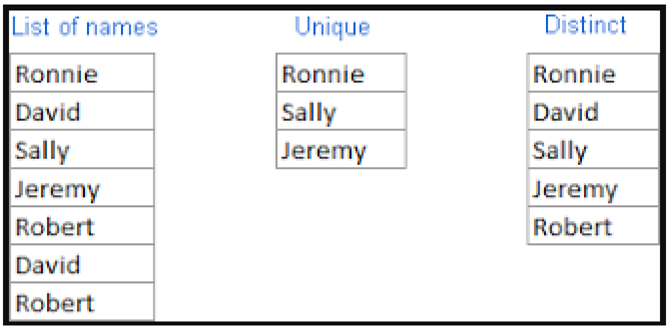


G2 =UNIQUE(A2:B13)

	A	B	C	D	E	F	G	H
1	Region	Sales Person	Product	Month	Sales		Region	Sales Person
2	North	Buchanan	Beverages	Jan	$468		North	Buchanan
3	North	Davolio	Beverages	Jan	$944		North	Davolio
4	North	Buchanan	Produce	Jan	$423			
5	North	Davolio	Produce	Jan	$822			
6	North	Buchanan	Beverages	Feb	$339			
7	North	Davolio	Beverages	Feb	$579			
8	North	Buchanan	Produce	Feb	$787			
9	North	Davolio	Produce	Feb	$425			
10	North	Buchanan	Beverages	Mar	$545			
11	North	Davolio	Beverages	Mar	$971			
12	North	Buchanan	Produce	Mar	$835			
13	North	Davolio	Produce	Mar	$517			

"Distinct" means total number of different values regardless how many times it appears in the dataset. A name appears in the list multiple times is counted as 1 distinct count.

Whereas, the "Unique" value is total number of values that **only appear once**.



List of names	Unique	Distinct
Ronnie	Ronnie	Ronnie
David	Sally	David
Sally	Jeremy	Sally
Jeremy		Jeremy
Robert		Robert
David		
Robert		