Introducing Arrays

By Nate Moore, CPA, MBA, FACMPE

The last article in *RCM Advisor* concluded by stating that arrays can make Excel functions like LARGE and SMALL even more powerful. This article will introduce arrays. Arrays are Excel formulas that can make complex calculations. There are two types of array formulas: array formulas that calculate a single result, and array formulas that generate multiple results. This article will start on array formulas that analyze multiple cells to create a single result. The final example (using the SMALL function) will demonstrate an array with multiple results in a function.

Getting Started

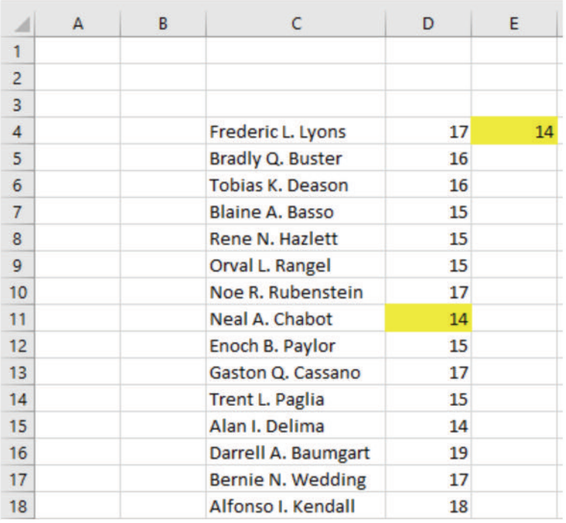
The way to identify an array formula is by the curly braces that surround a cell formula. The formula {=MIN(LEN(C4:C18))} is an array formula that calculates the length of each cell from C4:C18, and then calculates the minimum length in that range. The key is that you cannot enter the braces in the formula by typing them. Instead, when you finish typing a formula, use Ctrl+Shift+Enter and Excel automatically adds the braces. Sometimes array formulas are known as Ctrl+Shift+Enter or CSE formulas because of the keystrokes required to enter an array formula.

Figure 1 is an example of an array formula. Patient names are stored in cells C4:C18. Cell D4 has the formula =LEN(C4). The remaining cells in column D have a similar formula for each row. You could write the formulas in column D and then write a formula =MIN(D4:D18) to get the minimum length of patient names in column C, but an array formula is faster. Rather than write all the formulas in column D and then create another formula to take the minimum, the array formula {=MIN(LEN(C4:C18))} executes both the length calculations and the minimum calculation in a single step. Remember to type Ctrl+Shift+Enter to finish the formula. Use an example from your data to calculate a minimum length of cells and try an array formula.

Two Formulas at Once

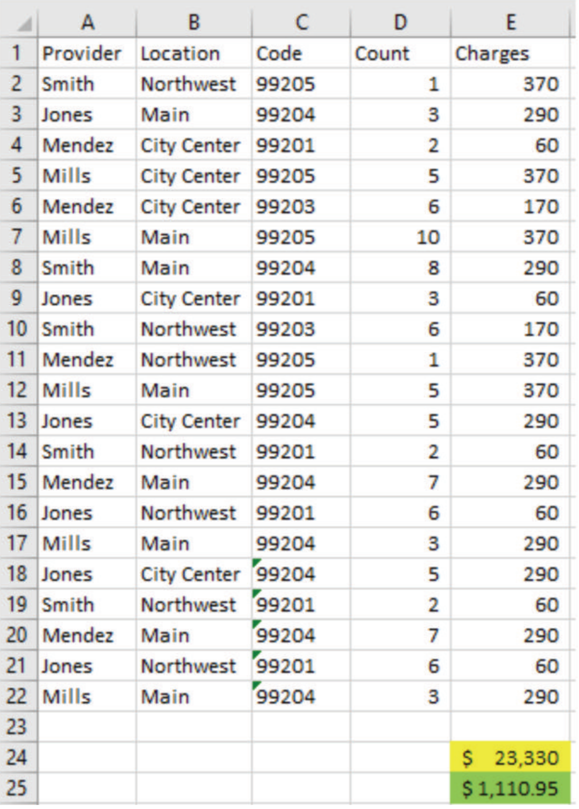
Array formulas can combine other functions into one formula. In Figure 2, assume your practice management system has exported the data and you need to calculate the total charges. Column D shows the number of units billed, and column E shows the billed charges. Like our first example, one way to calculate the total charges would be to multiply each row in column D by column E and store that result in column F. If you write another formula to sum the results in column F,

FIGURE 1



	A	B	C	D	E
1					
2					
3					
4			Frederic L. Lyons	17	14
5			Bradly Q. Buster	16	
6			Tobias K. Deason	16	
7			Blaine A. Basso	15	
8			Rene N. Hazlett	15	
9			Orval L. Rangel	15	
10			Noe R. Rubenstein	17	
11			Neal A. Chabot	14	
12			Enoch B. Paylor	15	
13			Gaston Q. Cassano	17	
14			Trent L. Paglia	15	
15			Alan I. Delima	14	
16			Darrell A. Baumgart	19	
17			Bernie N. Wedding	17	
18			Alfonso I. Kendall	18	

FIGURE 2



	A	B	C	D	E
1	Provider	Location	Code	Count	Charges
2	Smith	Northwest	99205	1	370
3	Jones	Main	99204	3	290
4	Mendez	City Center	99201	2	60
5	Mills	City Center	99205	5	370
6	Mendez	City Center	99203	6	170
7	Mills	Main	99205	10	370
8	Smith	Main	99204	8	290
9	Jones	City Center	99201	3	60
10	Smith	Northwest	99203	6	170
11	Mendez	Northwest	99205	1	370
12	Mills	Main	99205	5	370
13	Jones	City Center	99204	5	290
14	Smith	Northwest	99201	2	60
15	Mendez	Main	99204	7	290
16	Jones	Northwest	99201	6	60
17	Mills	Main	99204	3	290
18	Jones	City Center	99204	5	290
19	Smith	Northwest	99201	2	60
20	Mendez	Main	99204	7	290
21	Jones	Northwest	99201	6	60
22	Mills	Main	99204	3	290
23					
24					$ 23,330
25					$ 1,110.95

you can calculate the extended charge amount. Again, array formulas are faster. The formula in cell E24 (with yellow fill) is {=SUM(D2:D22*E2:E22)}. The array formula first multiplies column D by column E, and then sums the results.

Be careful with array formulas. When I copied the formula to include it in this article, Excel removed the braces. The result is in Figure 3. The array formula is still in cell E24, but the braces are missing. To fix the problem, simply edit the formula (either click inside the formula or use F2) and then finish the formula with Ctrl+Shift+Enter again to restore the curly braces and the array formula.

The array formula in cell E25 is {=AVERAGE(D2:D22*E2:E22)}. Working from the inside of the formula out, the array formula first multiplies columns D and E, and then takes the average of the multiplication. The $1,110.95 is the average *extended* charge of each CPT Code. It is not the average charge of each CPT code. You may have to build some testing columns to make sure you understand how the array formula makes its calculations, but arrays can save you lots of time and space once you are familiar with the syntax.

LARGE and SMALL

Now try using array formulas to put criteria in the LARGE and SMALL calculations from the last *RCM Advisor* article. The formula looks tricky, but with a little practice, you can add tremendous power to your array formulas. Look at Figure 4: The objective is to find the third next available appointment for Dr. Smith. The third next available is based on the 3 in cell C1. The array formula looks for the provider in cell B2. The array formula in cell A4 is =SMALL(IF(D4:D18=B2,C4:C18),3).

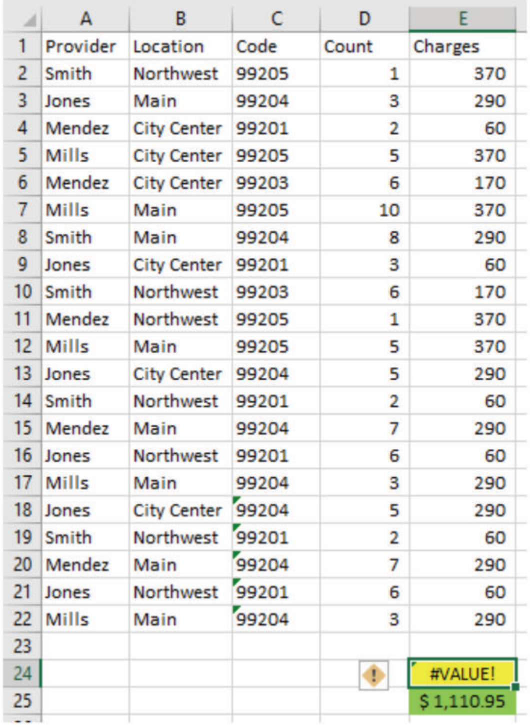
To understand the formula, remember that SMALL takes a range of cells and then counts for the smallest cell. The range of cells to search is an array. The 3 in the SMALL function means that Excel will take the third smallest value as the result of the formula. The array formula gets the range of cells with the IF function. If cells D4:D18 are equal to B2 (Smith), then the IF formula adds the respective appointment date in C4:C18 to the range of cells to search. Once the range of cells with Dr. Smith's appointments is generated, the array formula looks for the third smallest date. A traditional IF formula could not evaluate each cell in the range D4:D18 and store the corresponding cells in column C when the cell in column D is equal to Smith, but array formulas can.

There are Excel videos at mooresolutionsinc.com that introduce arrays and then use arrays to put criteria in a SMALL example:

mooresolutionsinc.com/arrayspart1
mooresolutionsinc.com/arrayspart2

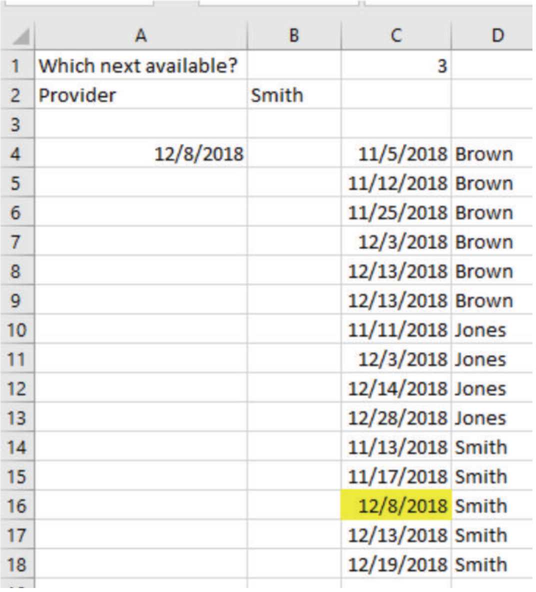
Nate Moore, CPA, MBA, FACMPE, writes custom SQL Server code to mine practice management data for analysis in Excel, web pages, and via email.

FIGURE 3



	A	B	C	D	E
1	Provider	Location	Code	Count	Charges
2	Smith	Northwest	99205	1	370
3	Jones	Main	99204	3	290
4	Mendez	City Center	99201	2	60
5	Mills	City Center	99205	5	370
6	Mendez	City Center	99203	6	170
7	Mills	Main	99205	10	370
8	Smith	Main	99204	8	290
9	Jones	City Center	99201	3	60
10	Smith	Northwest	99203	6	170
11	Mendez	Northwest	99205	1	370
12	Mills	Main	99205	5	370
13	Jones	City Center	99204	5	290
14	Smith	Northwest	99201	2	60
15	Mendez	Main	99204	7	290
16	Jones	Northwest	99201	6	60
17	Mills	Main	99204	3	290
18	Jones	City Center	99204	5	290
19	Smith	Northwest	99201	2	60
20	Mendez	Main	99204	7	290
21	Jones	Northwest	99201	6	60
22	Mills	Main	99204	3	290
23					
24				!	#VALUE!
25					$ 1,110.95

FIGURE 4



	A	B	C	D
1	Which next available?		3	
2	Provider	Smith		
3				
4	12/8/2018		11/5/2018	Brown
5			11/12/2018	Brown
6			11/25/2018	Brown
7			12/3/2018	Brown
8			12/13/2018	Brown
9			12/13/2018	Brown
10			11/11/2018	Jones
11			12/3/2018	Jones
12			12/14/2018	Jones
13			12/28/2018	Jones
14			11/13/2018	Smith
15			11/17/2018	Smith
16			12/8/2018	Smith
17			12/13/2018	Smith
18			12/19/2018	Smith

Nate's second book, Better Data, Better Decisions – The SQL: Business Intelligence for Medical Practices, *was recently published by MGMA. His free Excel videos have been viewed over 2 million times and are available at mooresolutionsinc.com. Like PivotTableGuy on Facebook or follow PivotTableGuy on Twitter to be notified each time an Excel Video is released. To get hands-on training with Nate, attend the HBMA Annual Conference.*