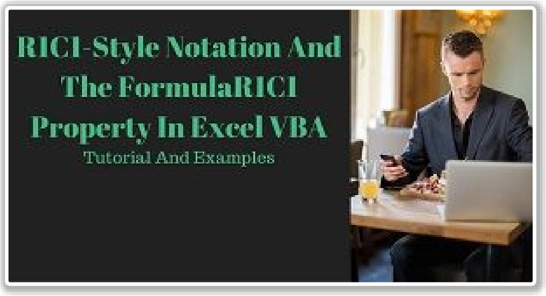
PS
Power Spreadsheets
Become an Excel Power User

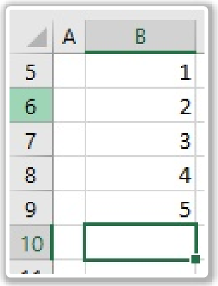
R1C1-Style Notation And The FormulaR1C1 Property In Excel VBA: Tutorial And Examples

By Jorge A. Gomez



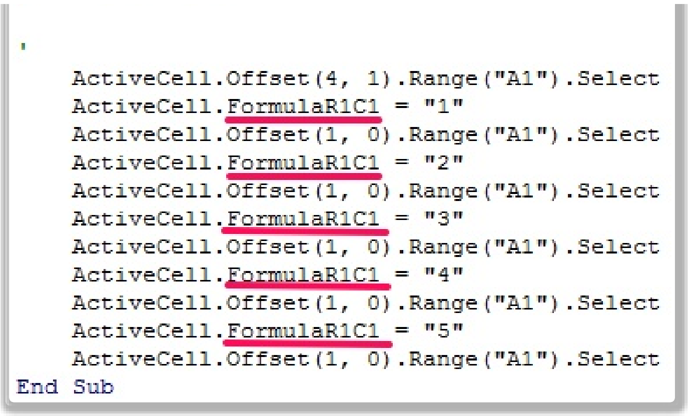
If you've used the macro recorder, you're probably familiar with both R1C1-style notation and the FormulaR1C1 property. This is because, as explained at Stack Overflow, the macro recorder constantly uses FormulaR1C1.

For example, even though you normally use the Range.Value property for purposes of entering a value in a cell, the macro recorder uses the Range.FormulaR1C1 property for those same purposes. To give you an idea, I used the macro recorder for purposes of creating the following sample macro (Enter_Value_FormulaR1C1). I recorded my actions while **(i)** entering numbers 1 through 5 in cells B5 to B9 and **(ii)** selecting cell B10 at the end.



	A	B
5		1
6		2
7		3
8		4
9		5
10		

Notice how the macro recorder uses the FormulaR1C1 property every single time.



```
'
    ActiveCell.Offset(4, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "1"
    ActiveCell.Offset(1, 0).Range("A1").Select
    ActiveCell.FormulaR1C1 = "2"
    ActiveCell.Offset(1, 0).Range("A1").Select
    ActiveCell.FormulaR1C1 = "3"
    ActiveCell.Offset(1, 0).Range("A1").Select
    ActiveCell.FormulaR1C1 = "4"
    ActiveCell.Offset(1, 0).Range("A1").Select
    ActiveCell.FormulaR1C1 = "5"
    ActiveCell.Offset(1, 0).Range("A1").Select
End Sub
```

As explained in *Excel 2016 Power Programming with VBA* and by Bob Phillips at excelforum.com, you normally use the Range.Value property for purposes of entering a value in a cell. In practice, using FormulaR1C1 "produces the same result".

However, my focus on this particular VBA tutorial isn't comparing the Range.FormulaR1C1 property with the Range.Value property. **My focus is the Range.FormulaR1C1 property itself and the R1C1-style notation**.

More particularly, **my purpose with this blog post is to provide you with all the information you need to understand and use both the R1C1-style notation and the Range.FormulaR1C1 property**.

You might be wondering whether you should take the time to learn about these 2 topics.

If that's the case, you can refer to the section below where I explain why R1C1-style references and the Range.FormulaR1C1 property are useful. For the moment, and to make it short, my opinion is that **you should take the time to learn and understand both the R1C1-style notation and the FormulaR1C1 property because**, as explained by Excel authorities Bill Jelen (Mr. Excel) and Tracy Syrstad in *Excel 2016 VBA and Macros*:

> Taking 30 minutes to understand R1C1 will make every macro you write for the rest of your life easier to code.

The following table of contents lists the main sections of this Excel tutorial:

Table of Contents [hide]

Now, let's start by taking a look at…

R1C1-Style And A1-Style Notation: A Basic Introduction

In order to understand the rest of this Excel tutorial, and how the FormulaR1C1 property may help you when working with VBA, having a good understanding of R1C1 notation is useful. I provide an introduction to the R1C1 notation in this section.

First, let me explain what I mean by "notation":

When working with Excel formulas, **notation is (in broad terms) the system you use to represent a cell reference**. When referring to a cell in Excel, you usually create the reference by taking into consideration the following 2 items:

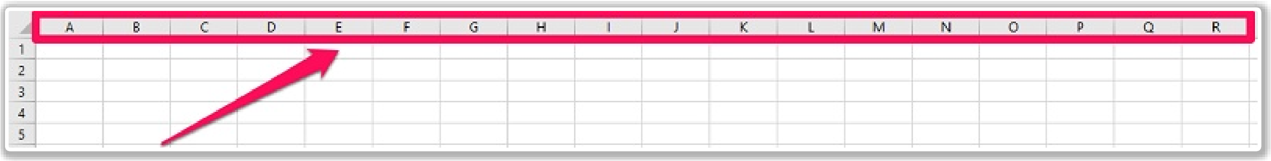
- Column.
- Row.

If you've been working with Excel for a while, you're probably quite familiar with the most common notation:

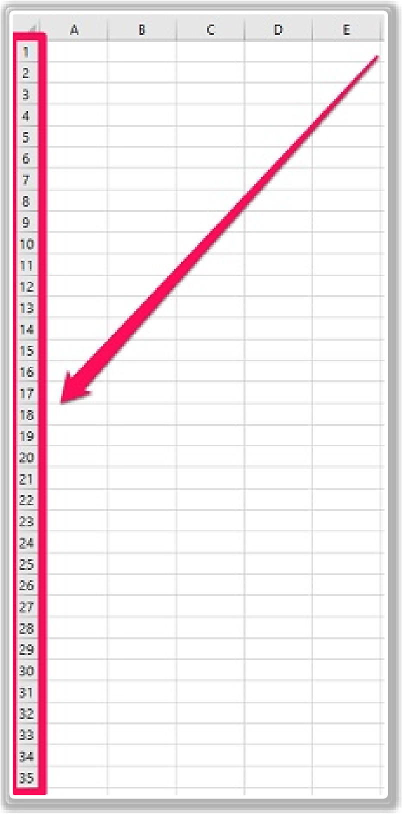
A1-Style Notation

The A1 reference style is **Excel's default style notation**.

- **Refer to the column by using the letter** that appears in the column heading. In recent Excel versions, this letter can be from A to XFD.



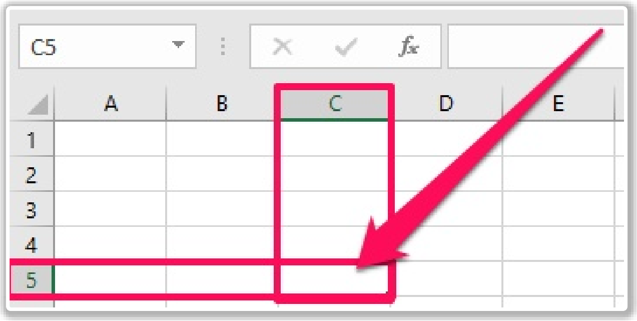
- **Refer to the row by using the number** that appears in the row heading. In recent Excel versions, this number ranges from 1 to 1,048,576.



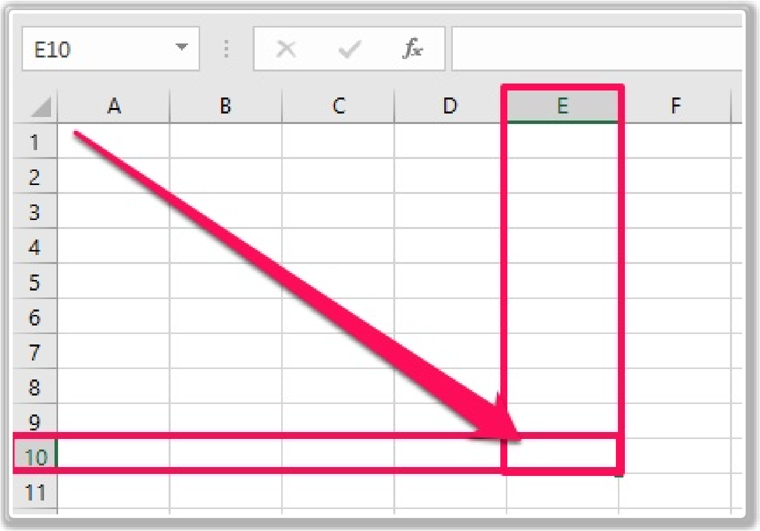
When using the A1-style notation, you create a **cell reference by concatenating the column letter and the row number**. For example:

- "A1" refers to the first cell in the worksheet, where column A and row 1 intersect.

- “C5” refers to the cell at the intersection of column C and row 5.



- “E10” makes reference to the cell where column E and row 10 intersect.



- …

- And so on. You probably get the idea.

In other words, **A1-style notation is simply the cell reference style you’re used to working with** in Excel.

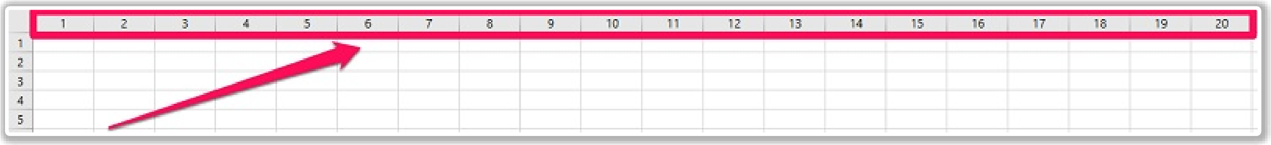
R1C1-Style Notation

R1C1-style notation is the alternative reference style to the A1-style notation I explain in the previous section.

The **main difference between the A1 and R1C1 notations is the way in which columns are identified**. Here's what I mean:

- When you're using the A1-style notation, columns are identified by letters. I explain this in the previous section.

- When you're working **with R1C1-style references, columns are identified by numbers**. In recent Excel versions, this number can be from 1 to 16,384.



What's the bottom line?

Well, as explained by Microsoft, in the R1C1-style notation:

> Both the rows and the columns on the worksheet are numbered.

A **second, also important, difference** between the A1 and R1C1 styles is the way in which you identify rows and columns when building cell references. As I mention above, when you're working with the A1 system, you create a reference by concatenating **(i)** the column identifier and **(ii)** the row identifier.

When you're working the R1C1-style references, **2 things change**:

- **#1:** The order in which you concatenate the rows and columns. Therefore:

  If you're working with A1-style references, **(i)** the column goes first and **(ii)** the row goes second.

  If you're using the R1C1 style, you concatenate the rows and columns in the opposite order. That is: **(i)** the row goes first and **(ii)** the column goes second.

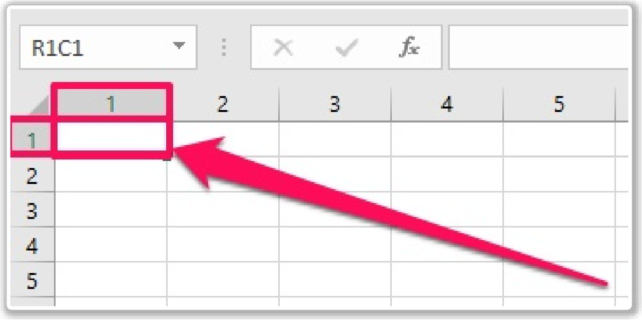
- **#2:** The way in which you identify the relevant rows and columns. What I mean is the following:

  If you're using the A1 style, you concatenate the column letter and the row number. This is what I

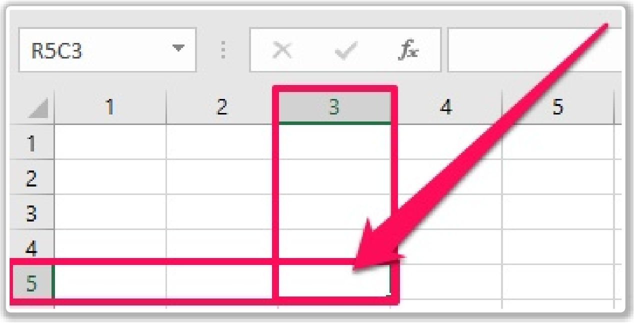
If you're working with the R1C1-style notation: **(i)** the column number is preceded by the letter "C" and **(ii)** the row number is preceded by the letter "R".

Let's see how this looks like in practice:

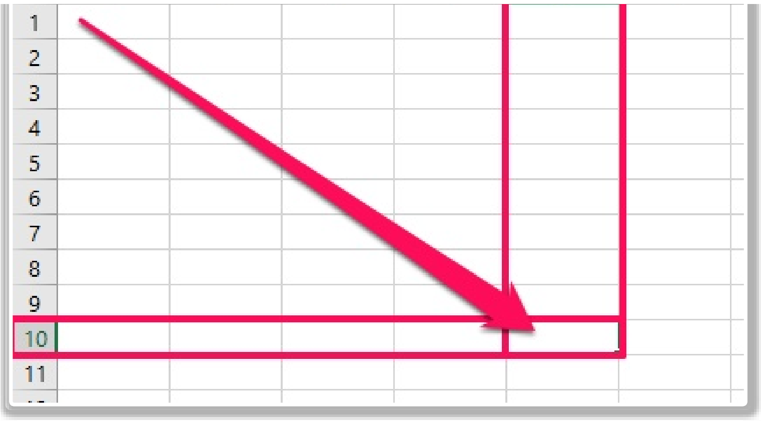
- "R1C1" refers to the cell at the intersection of the first column and the first row. In A1-style notation (as I explain above) you refer to this cell as A1.



- "R5C3" makes reference to the cell where the fifth row and the third column intersect. In the example that I provide in the previous section (using the A1 style), you call this cell C5.



- "R10C5" is the cell where the tenth row and the fifth column intersect. When working with A1-style references, this is cell E10.



Now that you understand the R1C1-style notation, you may be wondering things such as:

- *Why is the R1C1 style useful?*
- *Why is the focus of this VBA tutorial the FormulaR1C1 property instead of the Range.Formula property?*
- *Why should you understand R1C1-style references and the FormulaR1C1 property when working with VBA?*

To answer these (and similar questions) you may have, let's take a look at:

R1C1-Style References And The FormulaR1C1 Property: Why Are They Important And Useful

In Chapter 5 of *Excel 2016 VBA and Macros*, Excel authorities Bill Jelen (Mr. Excel) and Tracy Syrstad provide a very good introduction to **the historical background of the A1 and R1C1 referencing styles**. The short story, as explained at ExcelMate, is roughly as follows:

- **#1:** Back in the early days of spreadsheets, VisiCalc (the first spreadsheet program) introduced the A1 notation. Lotus 1-2-3, a very popular spreadsheet program in the 1980s, followed suit and also used A1-style references.
- **#2:** Microsoft's Multiplan (an early spreadsheet program from Microsoft) used the R1C1-style notation.
- **#3:** Lotus 1-2-3 was very popular during the 1980s.

> Officially (…), Microsoft supports both styles of addressing.

For most practical purposes, nobody (or virtually nobody) uses the R1C1-style of referencing when working with Excel. However, when working with VBA, this isn't the case.

For starters, as I show at the beginning of this post, the Macro Recorder constantly uses the Range.FormulaR1C1 property. Therefore, having a good understanding of **R1C1 references allows you to read the code that the Macro Recorder creates**.

However, this isn't the main strength of the R1C1-style notation. Here's the deal:

When you're working with Visual Basic for Applications, **R1C1-style references allow you to (for most purposes) create more efficient and powerful** VBA applications. Additionally, **if you want to be able to use certain features, you must use the R1C1-style notation**. In the words of Mr. Excel and Tracy Syrstad (in *Excel 2016 VBA and Macros*):

> I have to give Microsoft credit. R1C1-style formulas, you'll grow to understand, are actually more efficient, especially when you are dealing with writing formulas in VBA. Using R1C1-style addressing enables you to write more efficient code. Plus, there are some features such as setting up array formulas that require you to enter a formula in R1C1 style.

In a similar tone, Excel power user Puneet Gogia, states that:

> When it comes to VBA, R1C1 works like (…) magic.

In fact, as explained in this thread at ozgrid.com:

> Excel still uses R1C1 style under the covers.

At Chandoo.org, you can find some additional reasons why (as a general matter) you understanding the R1C1-style notation is helpful. This includes, for example, the fact that the **R1C1 style may help you when working with** VBA loops.

All of the above **doesn't mean that you must always use R1c1-style notation** when working with Visual Basic for Applications. There are several Excel experts and power users, including Jon Peltier, Dick by each of these Excel authorities.

Furthermore, as I explain below, **there are cases where you can rely on the A1-style notation** while working with VBA.

Given this, you might still wonder whether the R1C1-style notation is really more efficient than the A1 style when working with Visual Basic for Applications.

Fortunately, you don't need to take my (or Mr. Excel's) word. Let's take a closer look at the R1C1-style notation to see some evidence of this:

R1C1-Style Notation: How Are Cell References Created

In a previous section, I provide an introductory explanation of how you can create cell references that use the R1C1 style. At the most basic level, you can **refer to any cell by using the R1C1-style notation by concatenating the following 4 elements**:

- **Element #1:** The letter "R".

- **Element #2:** The reference to the number of the row where the cell is.

- **Element #3:** The letter "C".

- **Element #4:** The reference to the number of the cell's column.

I provide some introductory examples of cell references above.

When working with Excel, you can encounter (and use) **3 different types of cell references**. This applies regardless of whether you're using the A1 or the R1C1-style notation. These 3 types of cell references, as explained at office.com, are:

- **Type #1:** Relative references.

  Relative references **change as you copy** a formula from one cell to another.

  When you're working with A1-style references, relative references are **the default**. For example, when creating a relative reference to refer to the first cell of a worksheet using A1-style notation, you simply write "A1".

  Absolute references **don't change as you copy** a formula from one cell to another.

  As explained in the *Excel 2016 Bible*, absolute references **use 2 dollar signs ($)**. One of the dollar signs ($) goes before the column identifier. The other dollar sign ($) is placed before the row identifier. Therefore, to create an absolute reference to the first cell of a worksheet using A1-style notation, you type "A1".

- **Type #3:** Mixed references.

  As implied by their name, mixed references are **a mix of relative and absolute references**. Therefore, if you're using mixed references, **either the row or the column is absolute** (doesn't change as you copy the formula from one cell to another) and **the other is relative** (changes as you copy the formula).

  Mixed references **use a single dollar sign ($)**. This dollar sign is placed prior to the identifier of the item (row or column) that is absolute. For example, if you're creating a mixed reference to the first cell of a worksheet: **(i)** "$A1" results in a mixed reference where the column (A) is absolute and the row (1) is relative, and **(ii)** "A$1" results in a reference where the column (A) is relative and the row (1) is absolute.

If you're interested in learning more about relative, absolute and mixed cell references when using the A1-style notation, the following resources may be of interest to you:

- *Switch between relative, absolute, and mixed references* at office.com.

- *Cell References* at Excel Easy.

- *Relative And Absolute Cell References* at GCFLearnFree.org.

Since the focus of this VBA tutorial is the R1C1-style notation, let's take a look at how you create relative, absolute and mixed references with it:

Relative Reference With R1C1-Style Notation

It might sound crazy, but:

At its most basic level, the way in which you build relative references using the R1C1-style notation (which I explain in this section) is the main reason why R1C1-style references allow you to make your VBA code more efficient. So, even though the R1C1-style reference examples I provide below may look relatively

Note also that, despite the usefulness of the R1C1-style notation, **there are some issues you should keep in mind when creating macros that rely on such notation**. I provide a thorough explanation of this topic towards the end of this blog post.

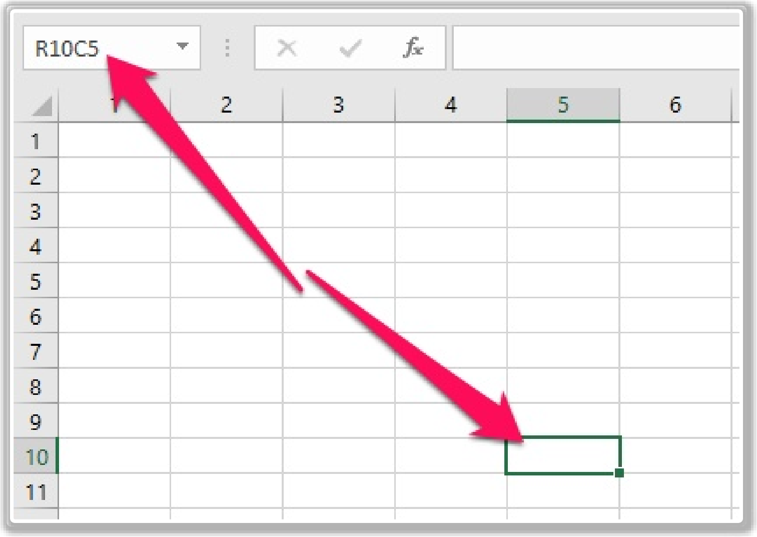
Relative references are the default in the R1C1-style notation.

R1C1-style **relative references have square brackets ([ ]) around the numbers of the rows and columns**.

However, relative R1C1-style references are subject to additional rules. To understand most of this, it **may help if you think of relative references in the R1C1-style notation being built in the following 3 steps**:

Step #1: Start On The Active Cell

Imagine that you're currently in cell R10C5. When using A1-style notation, this is cell E10.



R10C5

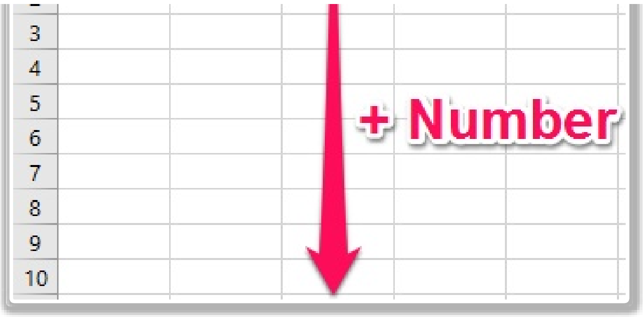
This cell is the base (or reference) for the relative reference you're building.

Step #2: Move A Certain Number Of Rows Up Or Down

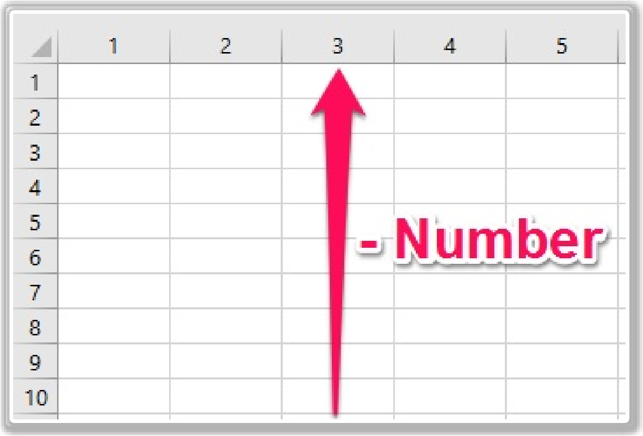
You **specify the row of the cell you're referring to by moving a certain number of rows up or down**. For these purposes you count from the cell you're working on (the current active cell).

The number of rows you're moving up or down determines the number that goes after the letter "R" in an R1C1-style reference. More precisely:

- If you use a **positive number, you're moving down** the worksheet.



- If you use a **negative number, you're moving up** the worksheet.



- If you use **no number at all, you move neither** up nor down.

For example, if you're in cell R10C5 and you want to refer to the cell immediately below it (R11C5), the first portion of the cell reference is "R[1]".

However, if you're referring to the cell immediately above R10C5, the first portion of the cell reference is "R[-1]".

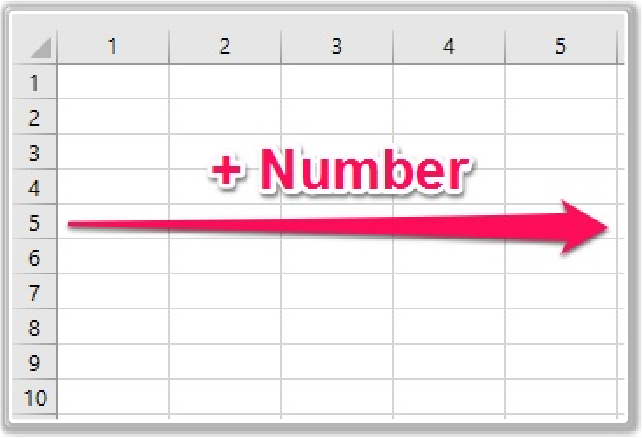
Finally, if you're referring to the cell itself (creating a circular reference), the first portion of the cell reference is simply "R".

Step #3: Move A Certain Number Of Columns To The Right Or To The Left

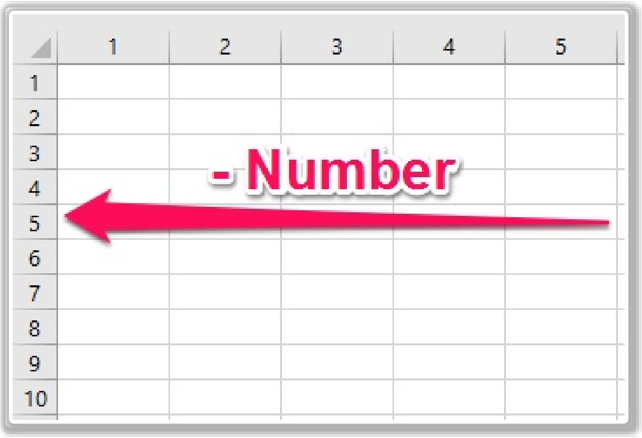
The rules for specifying the column to which you're referring to are a reflection of the rules I explain above to refer to the row number. Here's what I mean:

- The number of columns you move to the right or to the left determines the number that goes after the letter “C” in the R1C1-style reference.

- **Positive numbers mean you’re moving to the right** along the worksheet.



- **Negative numbers mean you’re moving to the left** along the worksheet.



- **No number means you’re staying on the same** column.

Continuing with the same example as in the previous step #2, if you’re in cell R10C5 and want to refer to the cell to the right, the second portion of the cell reference is “C[1]”. To refer to the cell to the left of R10C5, the second portion of the reference is “C[-1]”. To refer to the cell itself (and create a circular reference), the first portion of the cell reference is “C” alone.

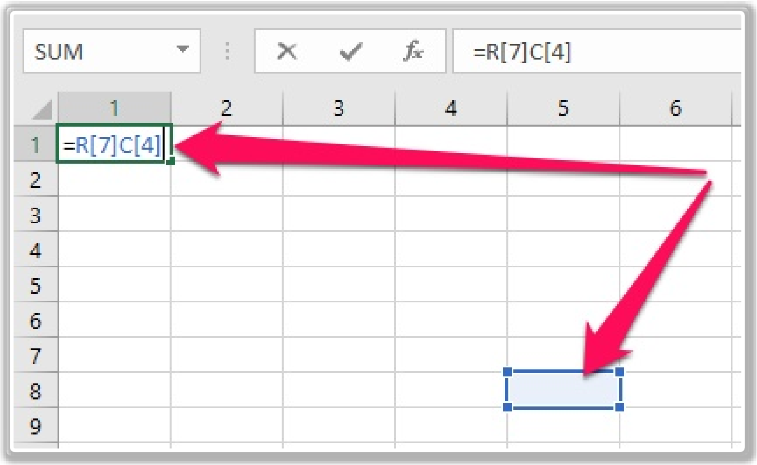
Let’s take a look at 4 examples of relative R1C1-style references:

Example #1 Of R1C1-Style Relative Reference

- 7 rows down; and

- 4 columns to the right.

Therefore, the reference is “R[7]C[4]”.



SUM

=R[7]C[4]

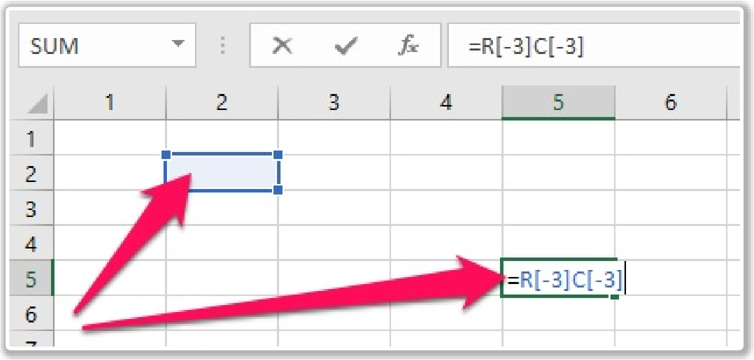
Example #2 Of R1C1-Style Relative Reference

If the current active cell is R5C5 (E5 in A1-style notation) and want to refer to cell R2C2 (B2 in A1-style notation), you move:

- 3 rows up; and

- 3 rows to the left.

As a consequence of this, the reference is “R[-3]C[-3]”.



SUM

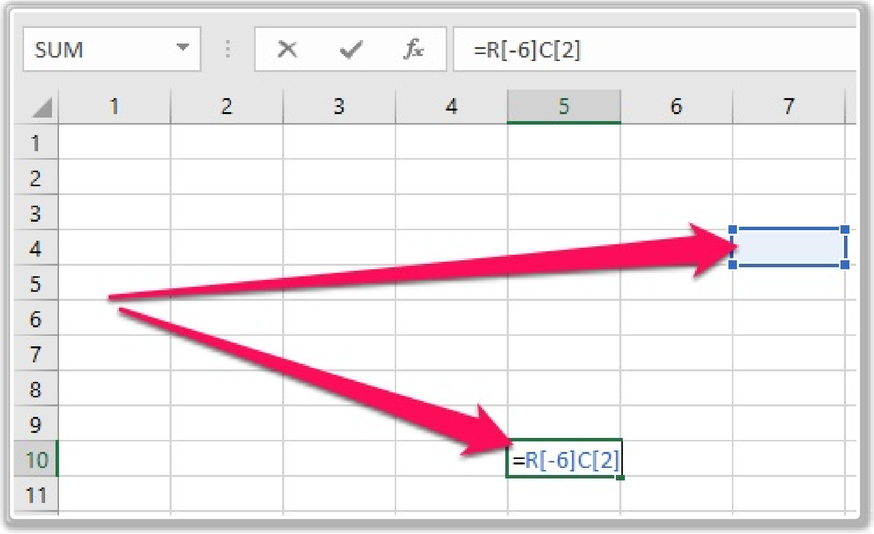
=R[-3]C[-3]

Example #3 Of R1C1-Style Relative Reference

If you’re in cell R10C5 (E10 in A1-style notation) and want to refer to cell R4C7 (G4 in A1-style notation), you move:

- 2 columns to the right.

Therefore, the reference is “R[6]C[2]”.


SUM

=R[-6]C[2]

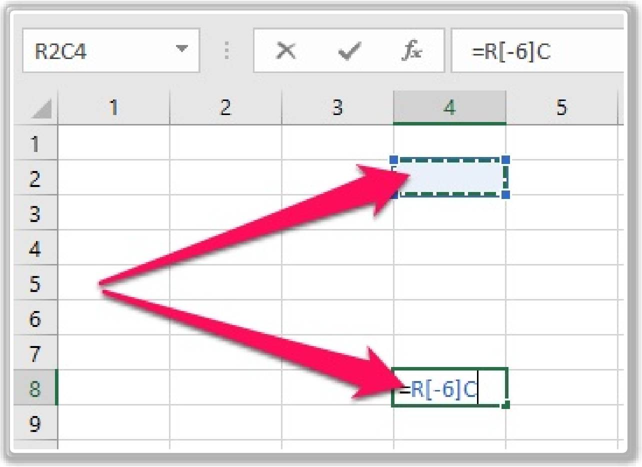
=R[-6]C[2]

Example #4 Of R1C1-Style Relative Reference

If you’re in cell R8C4 (cell D8 in A1-style notation) and want to refer to cell R2C4, you move:

- 6 rows up, and
- Stay in the same column.

Therefore, the reference is “R[-6]C”.


R2C4

=R[-6]C

=R[-6]C

You probably get the idea. Therefore, let’s move on to…

Absolute References With R1C1-Style Notation

This is the style that I use above when introducing the R1C1-style notation.

For example, when creating an absolute R1C1-style reference to the first cell in a worksheet, type “R1C1”.

Mixed References With R1C1-Style Notation

To build mixed references using the R1C1-style notation, **include the square brackets ([ ]) around the number of the item (row or column) you want to make relative**. In other words:

- To build a mixed R1C1-style reference where the row is relative and the column is absolute, surround the row number with square brackets ([ ]).

  Such a mixed reference looks roughly like “R[#]C#”.

- To create a mixed R1C1-style reference in which the row is absolute and the column is relative, wrap the column number with square brackets ([ ]).

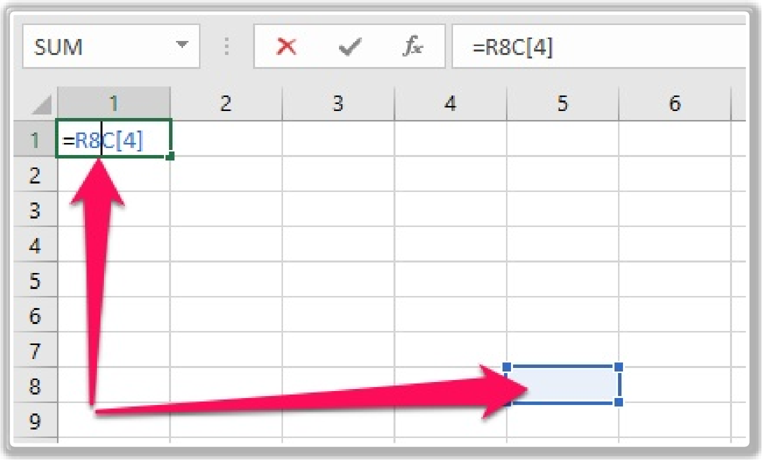
  In this case, the mixed references look roughly like “R#C[#]”.

Let’s go back to example #1 of a relative reference using the R1C1-style notation to see how a relative R1C1-style reference looks like:

In that situation, the active cell is R1C1 (A1 in A1-style notation). You’re referring to cell R8C5 (E8 in A1-style notation). The relative reference in such a situation is R[7]C[4].

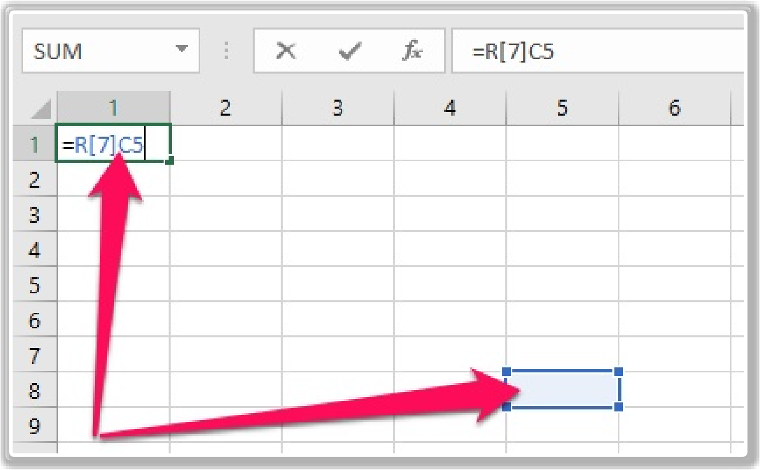
In such a case, you can build the following 2 mixed references:

- **Reference #1:** R8C[4]. In this case, the row is absolute and the column is relative.



=R8C[4]

- **Reference #2.** R[7]C5. In this reference, the row is relative and the column is absolute.



=R[7]C5

Referring To A Full Row Or Column Using R1C1-Style Notation

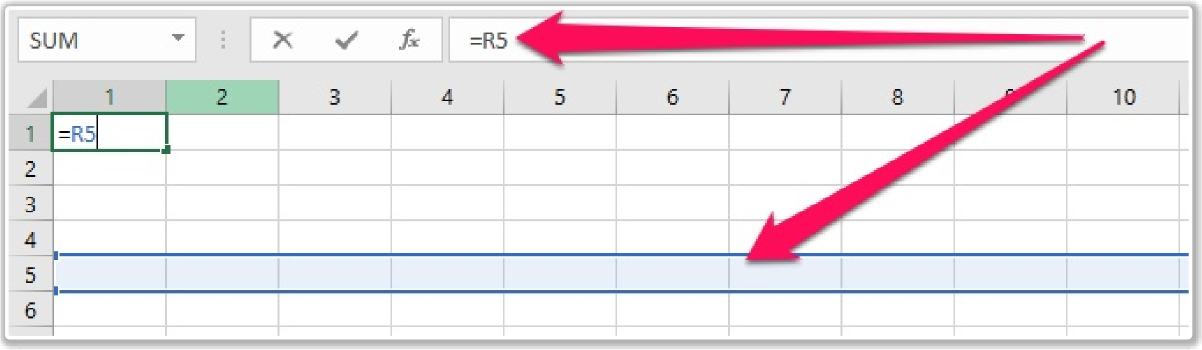
To create a reference to all the cells within a particular row or column, simply **omit the other item (row or column) of the reference**. What I mean is the following:

- To refer **to a full row, omit the column identifier**. Such a R1C1-style reference is of the form "R[#]" (if relative) or "R#" (if absolute).

- To refer **to a full column, omit the row identifier**. An R1C1-style reference is therefore of the form "C[#]" (if relative) or "C#" (if absolute).

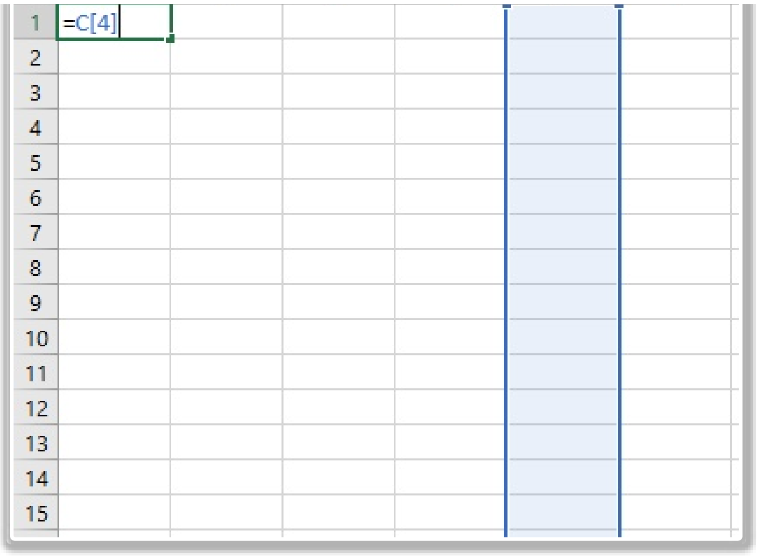
The rules regarding relative and absolute references, and the respective **use of square brackets ([ ]) continues to be the same** as I explain in the previous sections.

For example, R5 is an absolute reference to row 5.



=R5

Similarly, if the active cell is R1C1, C[4] refers to the full column located 4 columns to the right or R1C1 (column 5). This is a relative reference.


=C[4]

Now that you have a good grasp of the R1C1-style notation, let's take a look at the…

Range.FormulaR1C1 Property

One of the topics of focus of this VBA tutorial is the Range.FormulaR1C1 property. Let's take a look at its main characteristics:

Range.FormulaR1C1: Basic Description And Purpose

Range.FormulaR1C1 is a **read/write** property. Therefore, you can both **(i)** read the property or **(ii)** modify it.

The main purpose of the FormulaR1C1 property depends on whether you're reading or modifying the property. More precisely:

- If you're fetching the current setting of the property, Range.FormulaR1C1 **returns the formula (using R1C1-style notation)** of the relevant range.

- If you're modifying the value of the property, Range.FormulaR1C1 **sets the formula (using R1C1-style notation)** of the range you're working with.

The Range.FormulaR1C1 property **uses the language of the macro**. Therefore, the returned formula (when reading) or the set formula (when writing) are both in that language. If you're interested in dealing with **R1C1-style formulas in the language of the user, you may be interested in learning about the Range.FormulaR1C1Local** property (which I explain below).

Range.FormulaR1C1: Syntax

The **basic syntax** of the Range.FormulaR1C1 property is as follows:

“expression” is a variable representing a Range object.

Range.FormulaR1C1: Reading The Property

When you use the Range.FormulaR1C1 property for purposes of returning the formula of a particular range, the exact behavior of the property varies slightly depending on the contents of the relevant range. More precisely, as explained at the Microsoft Dev Center:

- If a cell **contains a constant**, Formula R1C1 returns the constant.

- If the cell is **empty**, FormulaR1C1 returns an empty string.

- If the cell **contains a formula**, FormulaR1C1 returns the formula **(i)** as a string and **(ii)** using the same format in which it’s displayed in the Formula Bar (even with the equal sign (=)).

Range.FormulaR1C1: Working With Ranges

If you use the Range.FormulaR1C1 property to set the formula of a multiple-cell range, **Excel fills all the cells of the relevant range** with the formula. Generally, as I show below, Excel **adjusts the cell references** automatically.

You can see how the FormulaR1C1 property works with multi-cell ranges in practice by taking a look at the practical examples below.

Range.FormulaR1C1: Setting A Date

If you use the Range.FormulaR1C1 property to set the value or formula of a particular cell to a date, Excel proceeds as follows:

- **Step #1:** It checks whether the cell you’re working with is formatted using a date or time number format.

- **Step #2:** If the condition is met, and the cell already uses a date or time number format, Excel leaves the format untouched.

  However, if the cell isn’t formatted using a date or time number format, Excel **sets the format of the cell to be the default short date number** format.

Range.FormulaR1C1Local Property And Language Considerations

> “ In general, you need not be concerned with the language in which you write your VBA code.

The reason for this, as explained by Kusleika and Alexander, is that Excel uses 2 object libraries:

- Excel’s object library.
- The VBA object library.

Excel always sets the English version of both libraries as the default. This is the case regardless of which language you use in Excel.

Working with formulas is, generally, one of the exceptions to the above rule. In other words, **if you’re working in a particular VBA application whose users may have different language settings, language compatibility may be an issue** you should consider.

Let’s take a closer look at this point:

Excel Formulas And Language Considerations

As a general matter, **Excel functions are localized**. As a consequence, the name of a particular function, such as IFERROR varies depending on the language in which Excel runs. Let me explain what I mean:

In this blog post, I explain several text functions such as LEFT, RIGHT, MID, LEN, FIND and SEARCH. These, however, are the names used by the English version of Excel. The **names of the functions change depending on the particular language** you’re using Excel in.

The following table shows the names of these particular functions in 10 different languages. I prepared the translation using the Excel-Translator.

#1	English	LEFT	RIGHT	MID	LEN	FIND	SEARCH
#2	Spanish	IZQUIERDA	DERECHA	EXTRAE	LARGO	ENCONTRAR	HALLAR
#3	German	LINKS	RECHTS	TEIL	LÄNGE	FINDEN	SUCHEN
#4	French	GAUCHE	DROITE	STXT	NBCAR	TROUVE	CHERCHE
#5	Dutch	LINKS	RECHTS	DEEL	LENGTE	VIND.ALLES	VIND.SPEC

#7	Portuguese (Brazil)	ESQUERDA	DIREITA	EXT.TEXTO	NUM.CARACT	PROCURAR	LOCALIZAR
#8	Polish	LEWY	PRAWY	FRAGMENT.TEKSTU	DŁ	ZNAJDŹ	SZUKAJ.TEKST
#9	Danish	VENSTRE	HØJRE	MIDT	LÆNGDE	FIND	SØG
#10	Turkish	SOLDAN	SAĞDAN	PARÇAAL	UZUNLUK	BUL	MBUL

Excel MVP Mourad Louha is an expert in this topic and has invested a substantial amount of time and resources in the development of the Excel-Translator. Mourad explains **the challenge when working with Excel functions in a multilingual setting** quite clearly as follows:

> If you send your Excel file to someone using a different language for Excel than you, the functions and formulas used in the workbook are automatically translated by Excel when opening the file. However, **the automatic translation usually does not work, if you directly insert foreign language formulas into your worksheet**.

When working with the FormulaR1C1 property, you fall within the second scenario (in bold). In other words:

If you're reading or setting a formula by **using the FormulaR1C1 VBA property, the functions are usually not automatically translated**.

The easiest way to deal with language compatibility concerns when working with the FormulaR1C1 property is to **use the local version of the FormulaR1C1 property**:

Range.FormulaR1C1Local Property

To understand why the LanguageR1C1Local may be helpful, let's go back to what I say above about the language used by the Range.FormulaR1C1. That is, the FormulaR1C1 property uses the language of the macro.

On the other hand, the Range.FormulaR1C1Local property, which I cover in this section, **uses the language of the user**.

For practical purposes, **other than this important aspect, all of the comments I make above regarding the Range.FormulaR1C1 property are applicable**. More precisely:

- You can use Range.FormulaR1C1Local to return (read) or set (write) the formula of a range using R1C1-style notation.

- The **basic syntax** of Range.FormulaR1C1Local is “expression.FormulaR1C1Local”.

  “expression” represents a Range object.

- The item returned by Range.FormulaR1C1Local (when used for reading purposes) depends on whether the cell **(i)** is empty, **(ii)** contains a constant, or **(iii)** contains a formula.

  I explain further details of this behavior above.

- When you use the Range.FormulaR1C1Local property to set a value or formula to a date: **(i)** Excel checks whether the cell is formatted using a date or time number format and **(ii)** sets the format to the default short date number format if the cell isn’t formatted as a date or time.

- When working with ranges, the remarks I make in connection with Range.FormulaR1C1 throughout this tutorial are also applicable to Range.FormulaR1C1Local.

In other words:

- When working with the FormulaR1C1 property and reading this tutorial, bear in mind the following:

  #1: Range.FormulaR1C1 uses the **language of the macro**.

  #2: Range.FormulaR1C1Local uses the **language of the user**.

- Other than the above and **for most practical purposes, Range.FormulaR1C1 and Range.FormulaR1C1Local behave and can be treated materially the same**.

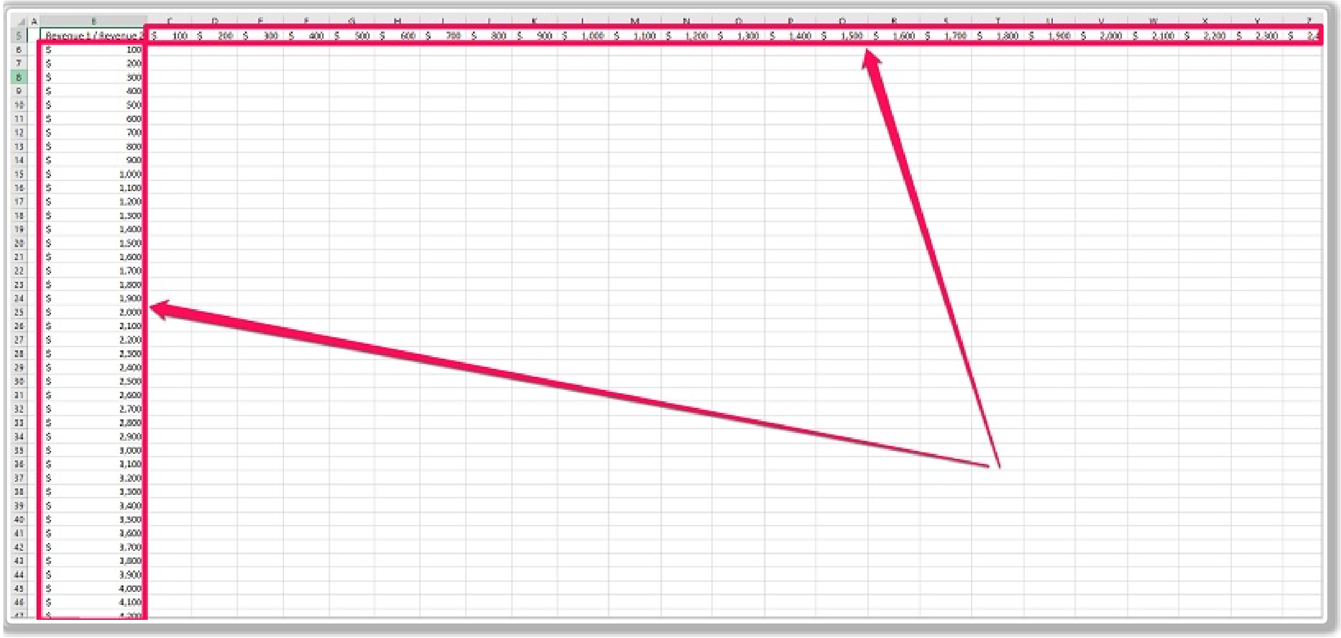
Range.FormulaR1C1 Property Example: Setting The Formula Of A Cell Range To Create A Table

I have prepared an Excel workbook that accompanies this tutorial. The workbook includes all the data and macros that I use (and show) throughout this blog post. You can **get immediate free access to this workbook (along all workbooks that accompany the other posts within Power Spreadsheets) by** clicking here.

For this example, let’s assume that you want to want to prepare a table that shows different combinations of the total revenue generated by 2 different revenue streams. The revenue generated by these streams is

- C5 (R5C3 in R1C1-style notation) to CX5 (R5C102 in R1C1-style notation).

- B6 (R6C2 in R1C1-style notation) to B105 (R105C2 in R1C1-style notation).



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
5		Revenue 1 / Revenue 2	$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,4
6		$ 100																								
7		$ 200																								
8		$ 300																								
9		$ 400																								
10		$ 500																								
11		$ 600																								
12		$ 700																								
13		$ 800																								
14		$ 900																								
15		$ 1,000																								
16		$ 1,100																								
17		$ 1,200																								
18		$ 1,300																								
19		$ 1,400																								
20		$ 1,500																								
21		$ 1,600																								
22		$ 1,700																								
23		$ 1,800																								
24		$ 1,900																								
25		$ 2,000																								
26		$ 2,100																								
27		$ 2,200																								
28		$ 2,300																								
29		$ 2,400																								
30		$ 2,500																								
31		$ 2,600																								
32		$ 2,700																								
33		$ 2,800																								
34		$ 2,900																								
35		$ 3,000																								
36		$ 3,100																								
37		$ 3,200																								
38		$ 3,300																								
39		$ 3,400																								
40		$ 3,500																								
41		$ 3,600																								
42		$ 3,700																								
43		$ 3,800																								
44		$ 3,900																								
45		$ 4,000																								
46		$ 4,100																								
47		$ 4,200																								

In order to fill the table and show the different total revenue combinations, you must add the following 2 numbers in each cell:

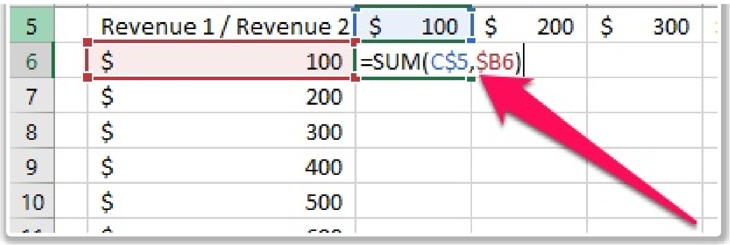
- The value at the top of the relevant column (in row 5); and

- The value at the beginning of the relevant row (in column B or 2).

Setting up the formula manually is relatively straightforward. You can do this in the following 2 easy steps:

- **Step #1:** Enter the relevant formula in the first cell of the table.

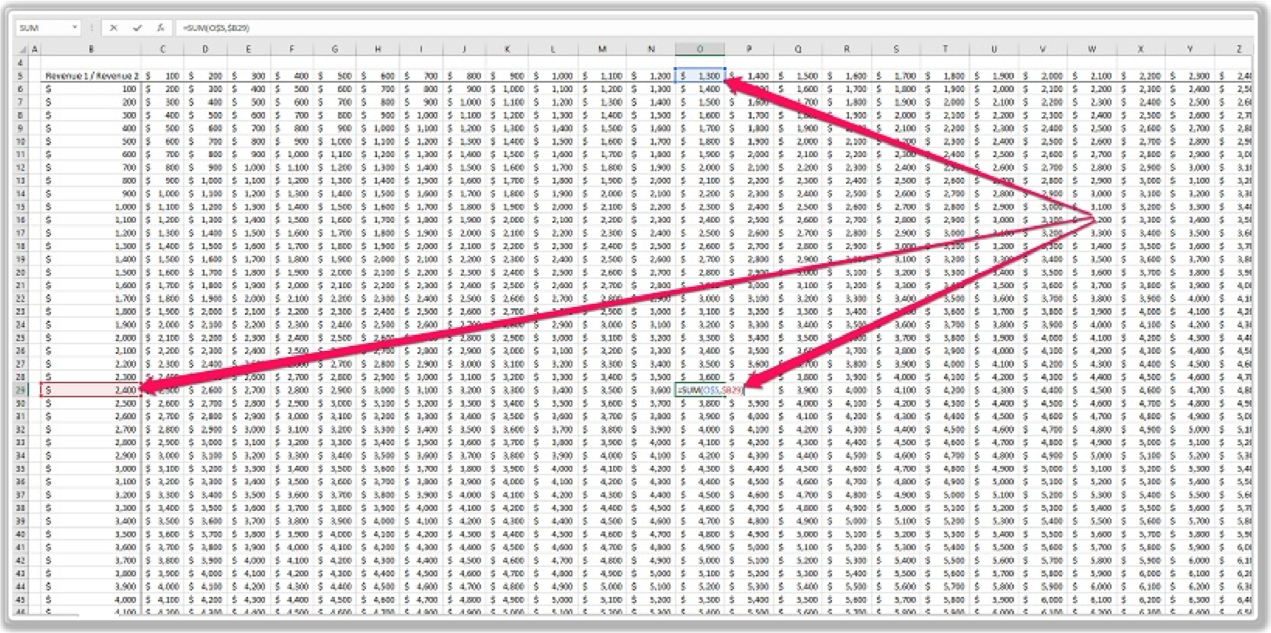
  In this particular case, the formula I enter is "=SUM(C$5,$B6)". Notice that the cell references are mixed: **(i)** in C$5, the row number is absolute and, therefore, isn't adjusted when the formula is copied to other cells, and **(ii)** in $B6, the column number is absolute and isn't adjusted as the formula is copied and pasted.



5	Revenue 1 / Revenue 2	$ 100	$ 200	$ 300
6	$ 100	=SUM(C$5,$B6)		
7	$ 200			
8	$ 300			
9	$ 400			
10	$ 500			

- **Step #2:** Copy the formula and paste it in all the relevant cells of the table.

  As expected, the table is filled with the appropriate formulas. Notice, for example, how the formula in cell O29 appropriately adds the cell at the top of the column (O5) and at the beginning of the row (B29).

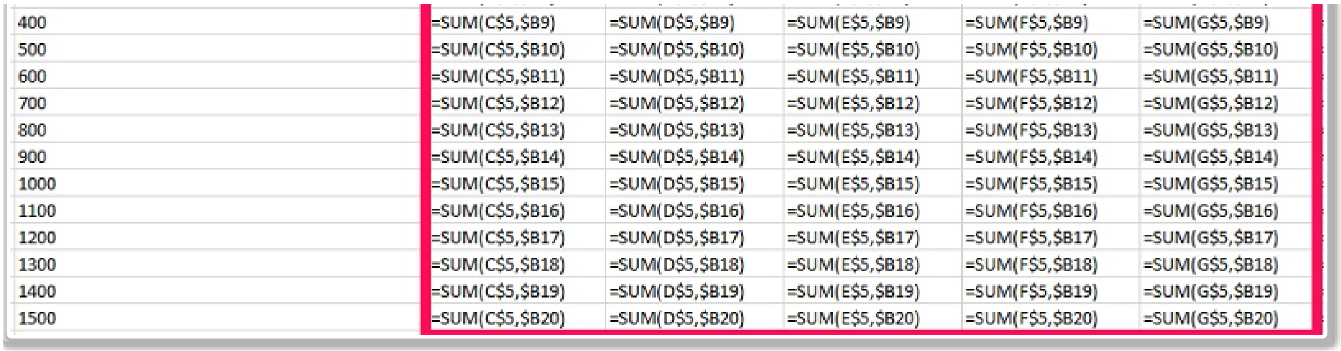


SUM

=SUM(O$5,$B29)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4																										
5		Revenue 1 / Revenue 2	$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,4
6		$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	[illegible]	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,5
7		$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	[illegible]	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,6
8		$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	[illegible]	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,7
9		$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	[illegible]	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,8
10		$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	[illegible]	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,9
11		$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	[illegible]	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,0
12		$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	[illegible]	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,1
13		$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	[illegible]	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,2
14		$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,3
15		$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	[illegible]	$ 3,100	$ 3,200	$ 3,300	$ 3,4
16		$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	[illegible]	[illegible]	$ 3,300	$ 3,400	$ 3,5
17		$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	[illegible]	[illegible]	$ 3,300	$ 3,400	$ 3,500	$ 3,6
18		$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	[illegible]	$ 3,200	[illegible]	$ 3,400	$ 3,500	$ 3,600	$ 3,7
19		$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	[illegible]	$ 3,100	$ 3,200	[illegible]	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,8
20		$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	[illegible]	[illegible]	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,9
21		$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	[illegible]	$ 3,000	$ 3,100	$ 3,200	$ 3,300	[illegible]	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,0
22		$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	[illegible]	[illegible]	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	[illegible]	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,1
23		$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	[illegible]	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	[illegible]	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,2
24		$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	[illegible]	[illegible]	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	[illegible]	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,3
25		$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	[illegible]	[illegible]	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	[illegible]	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,4
26		$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	[illegible]	[illegible]	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	[illegible]	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,5
27		$ 2,200	$ 2,300	$ 2,400	[illegible]	[illegible]	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	[illegible]	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,6
28		$ 2,300	[illegible]	[illegible]	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	[illegible]	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,7
29		$ 2,400	[illegible]	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	=SUM(O$5,$B29)	[illegible]	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,8
30		$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,9
31		$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,0
32		$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,1
33		$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,2
34		$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,3
35		$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,4
36		$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,5
37		$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,6
38		$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,7
39		$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,8
40		$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,9
41		$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,0
42		$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,1
43		$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,2
44		$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,3
45		$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,300	$ 6,4

The following screenshot shows some of the formulas within this worksheet. Notice how Excel (pretty much) rewrites a portion of the formula for each cell in order to **adjust the cell references**.



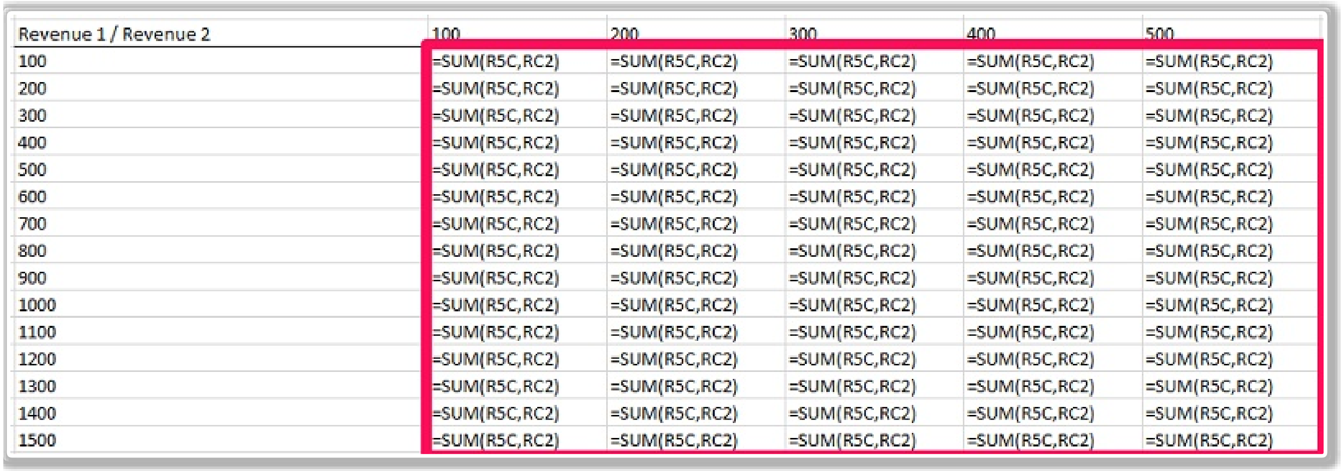
400	=SUM(C$5,$B9)	=SUM(D$5,$B9)	=SUM(E$5,$B9)	=SUM(F$5,$B9)	=SUM(G$5,$B9)
500	=SUM(C$5,$B10)	=SUM(D$5,$B10)	=SUM(E$5,$B10)	=SUM(F$5,$B10)	=SUM(G$5,$B10)
600	=SUM(C$5,$B11)	=SUM(D$5,$B11)	=SUM(E$5,$B11)	=SUM(F$5,$B11)	=SUM(G$5,$B11)
700	=SUM(C$5,$B12)	=SUM(D$5,$B12)	=SUM(E$5,$B12)	=SUM(F$5,$B12)	=SUM(G$5,$B12)
800	=SUM(C$5,$B13)	=SUM(D$5,$B13)	=SUM(E$5,$B13)	=SUM(F$5,$B13)	=SUM(G$5,$B13)
900	=SUM(C$5,$B14)	=SUM(D$5,$B14)	=SUM(E$5,$B14)	=SUM(F$5,$B14)	=SUM(G$5,$B14)
1000	=SUM(C$5,$B15)	=SUM(D$5,$B15)	=SUM(E$5,$B15)	=SUM(F$5,$B15)	=SUM(G$5,$B15)
1100	=SUM(C$5,$B16)	=SUM(D$5,$B16)	=SUM(E$5,$B16)	=SUM(F$5,$B16)	=SUM(G$5,$B16)
1200	=SUM(C$5,$B17)	=SUM(D$5,$B17)	=SUM(E$5,$B17)	=SUM(F$5,$B17)	=SUM(G$5,$B17)
1300	=SUM(C$5,$B18)	=SUM(D$5,$B18)	=SUM(E$5,$B18)	=SUM(F$5,$B18)	=SUM(G$5,$B18)
1400	=SUM(C$5,$B19)	=SUM(D$5,$B19)	=SUM(E$5,$B19)	=SUM(F$5,$B19)	=SUM(G$5,$B19)
1500	=SUM(C$5,$B20)	=SUM(D$5,$B20)	=SUM(E$5,$B20)	=SUM(F$5,$B20)	=SUM(G$5,$B20)

In the words of Mr. Excel and Tracy Syrstad (in *Excel 2016 VBA and Macros*):

> It seems intimidating to consider having a macro enter all these different formulas.

The reason for this, however, is because the formulas in the screenshots above use A1-style notation.

The following screenshot shows the same portion of the table we're working on as that above. The only difference is that, in the image below, R1C1-style notation is used.

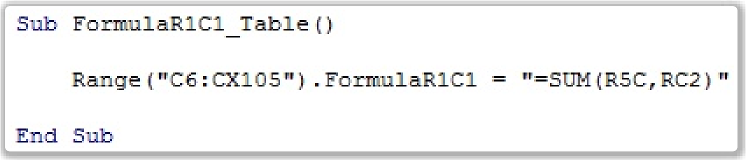


Revenue 1 / Revenue 2	100	200	300	400	500
100	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
200	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
300	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
400	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
500	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
600	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
700	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
800	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
900	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1000	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1100	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1200	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1300	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1400	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)
1500	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)	=SUM(R5C,RC2)

Notice that, **when using R1C1-style notation, all of the formulas in the table are exactly the same**.

As I explain above, you can use the Range.FormulaR1C1 property to enter such a formula in such a range.

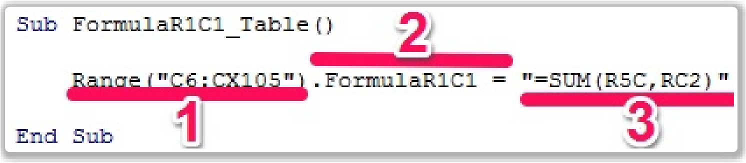
The following **sample macro (FormulaR1C1_Table) achieves this**:



```
Sub FormulaR1C1_Table()

    Range("C6:CX105").FormulaR1C1 = "=SUM(R5C,RC2)"

End Sub
```

> Range(“C6:CX105”).FormulaR1C1 = “=SUM(R5C,RC2)”

Let’s take a look at its 3 main components separately to understand how the macro works:



```
Sub FormulaR1C1_Table()

    Range("C6:CX105").FormulaR1C1 = "=SUM(R5C,RC2)"

End Sub
```

Item #1: Range(“C6:CX105”)

This makes reference to the Application.Range property. Within the syntax of the FormulaR1C1 property that I introduce above, this item takes the place of the expression that returns a Range object.

More precisely, this statement returns the range of cells from C6 to CX105 of the current active worksheet.

The fully qualified reference is “Application.ActiveSheet.Range(“C6:CX105”). However, as I explain on this VBA tutorial about Excel’s Object Model, you can simplify the reference by assuming the following default objects:

- The Application object.
- The Active Workbook.
- The Active Sheet.

Item #2: FormulaR1C1 =

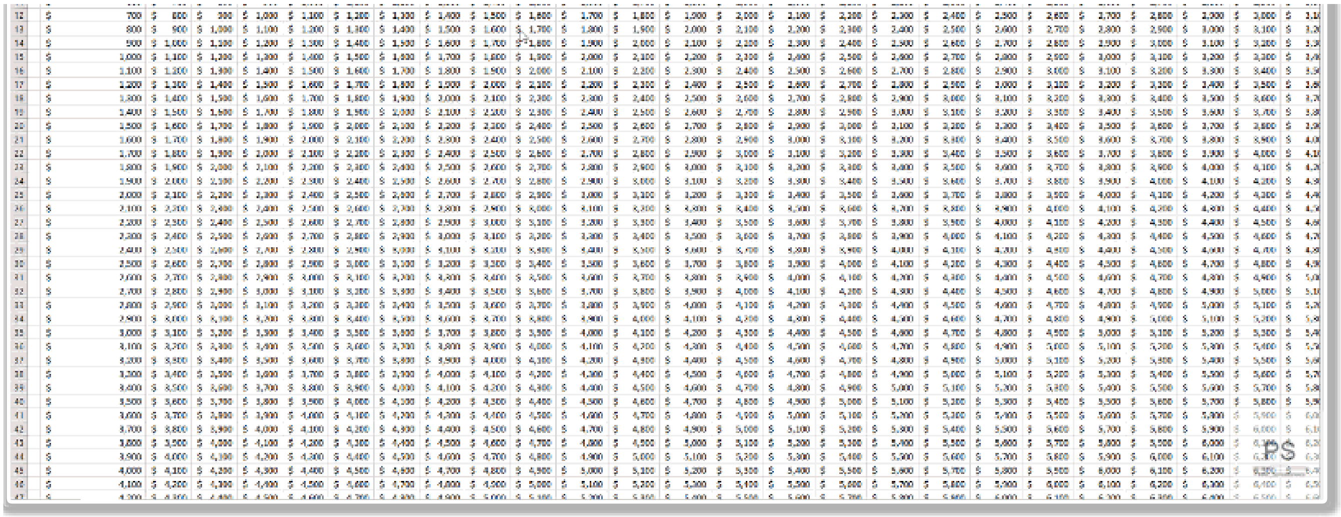
This item makes reference to **the FormulaR1C1 property**.

As I explain in the tutorial about VBA Object Properties, whenever you’re setting a property value, you use an equal sign (=) to separate the property name (FormulaR1C1) from the property value (item #3 below).

Item #3: “=SUM(R5C,RC2)”

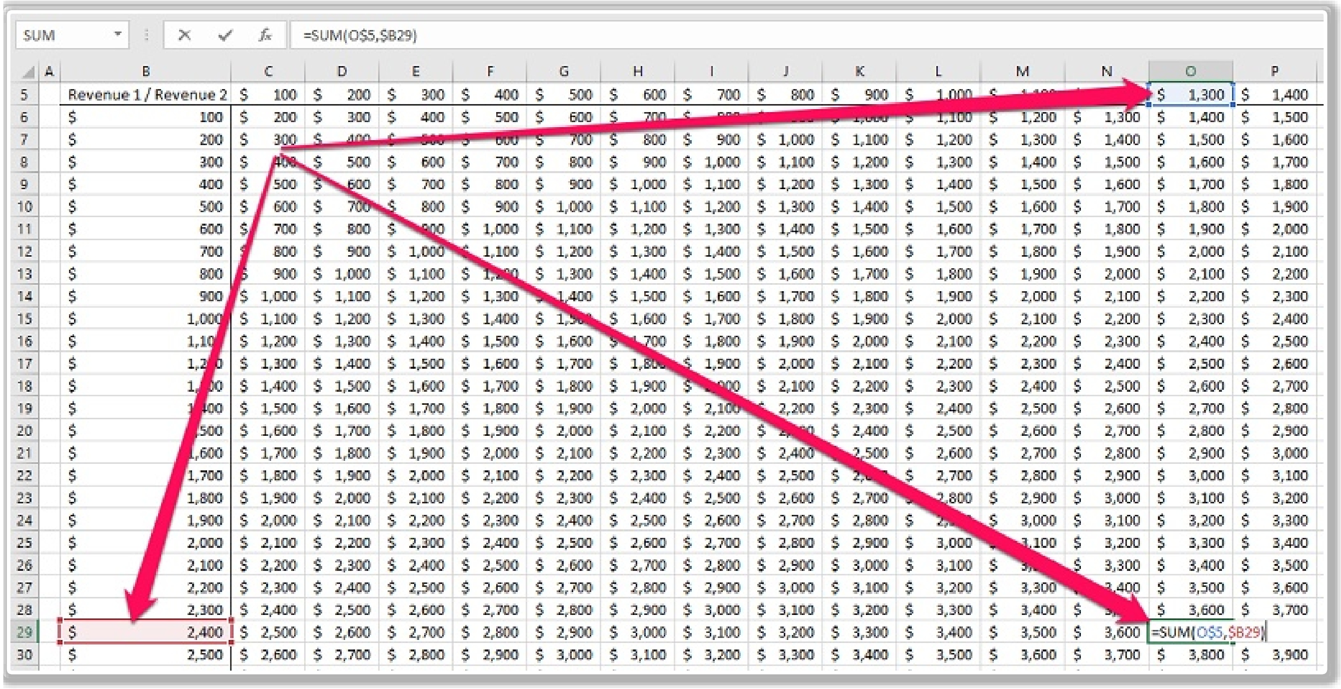
This item is the **value that is set for the FormulaR1C1 property of all the cells within the range** returned by item #1 above. This formula is the one I show in the screenshot above.

The following GIF shows what happens when I **execute the sample FormulaR1C1_Table macro**. Notice how, as expected, Excel appropriately fills all the cells within the table.



12	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,1
13	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,2
14	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,3
15	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,4
16	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,5
17	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,6
18	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,7
19	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,8
20	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,9
21	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,0
22	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,1
23	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,2
24	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,3
25	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,4
26	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,5
27	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,6
28	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,7
29	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,8
30	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,9
31	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,0
32	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,1
33	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,2
34	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,3
35	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,4
36	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,5
37	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,6
38	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,7
39	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,8
40	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,9
41	$ 3,600	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,0
42	$ 3,700	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,1
43	$ 3,800	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,2
44	$ 3,900	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,3
45	$ 4,000	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,300	$ 6,4
46	$ 4,100	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,300	$ 6,400	$ 6,5
47	$ 4,200	$ 4,300	$ 4,400	$ 4,500	$ 4,600	$ 4,700	$ 4,800	$ 4,900	$ 5,000	$ 5,100	$ 5,200	$ 5,300	$ 5,400	$ 5,500	$ 5,600	$ 5,700	$ 5,800	$ 5,900	$ 6,000	$ 6,100	$ 6,200	$ 6,300	$ 6,400	$ 6,500	$ 6,6

Let's make an additional check by reviewing the formula of cell O29. Notice that, in this case, the formula is exactly the same as that which resulted from the manual process I show above (=SUM(O$5,$B29)).



=SUM(O$5,$B29)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
5		Revenue 1 / Revenue 2	$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400
6		$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500
7		$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600
8		$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700
9		$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800
10		$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900
11		$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000
12		$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100
13		$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200
14		$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300
15		$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400
16		$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500
17		$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600
18		$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700
19		$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800
20		$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900
21		$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000
22		$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100
23		$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200
24		$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300
25		$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400
26		$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500
27		$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600
28		$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700
29		$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	=SUM(O$5,$B29)	
30		$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900

Considering all that I've said throughout this VBA tutorial regarding the advantages of using R1C1-style notation and the Range.FormulaR1C1 property when working with VBA, you may be surprised by what I explain in the following section:

Setting The Formula Of A Cell Range To Create A Table With The Range.Formula Property

Strictly speaking, you **can replicate the results of the sample FormulaR1C1_Table macro above by using A1-style notation and the Range.Formula property**.

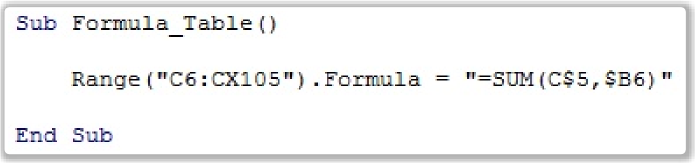
As explained by Excel expert Andrew Poulsom at MrExcel.com, the **Range.Formula property behaves very similarly to the Range.FormulaR1C1 property** I explain above. In other words, you can use the Formula property to either:

- Return (read) the formula (using A1-style notation) of the range you're working with.
- Set the formula (using A1-style notation) of the relevant range.

So, for purposes of this explanation, the Range.Formula and Range.FormulaR1C1 properties achieve the same purpose with 1 basic difference:

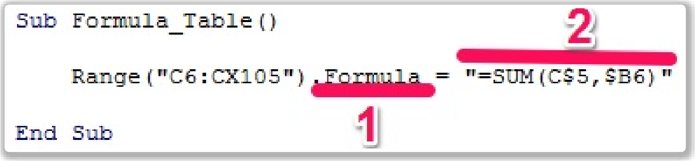
The **FormulaR1C1 property uses R1C1-style notation**. The **Formula property uses A1-style notation**.

The following **sample macro (Formula_Table) is the equivalent of the FormulaR1C1_Table example macro** above.



```
Sub Formula_Table()

    Range("C6:CX105").Formula = "=SUM(C$5,$B6)"

End Sub
```

The single statement in the Sub procedure is substantially the same. The **only 2 differences between the 2 macros** are the following:

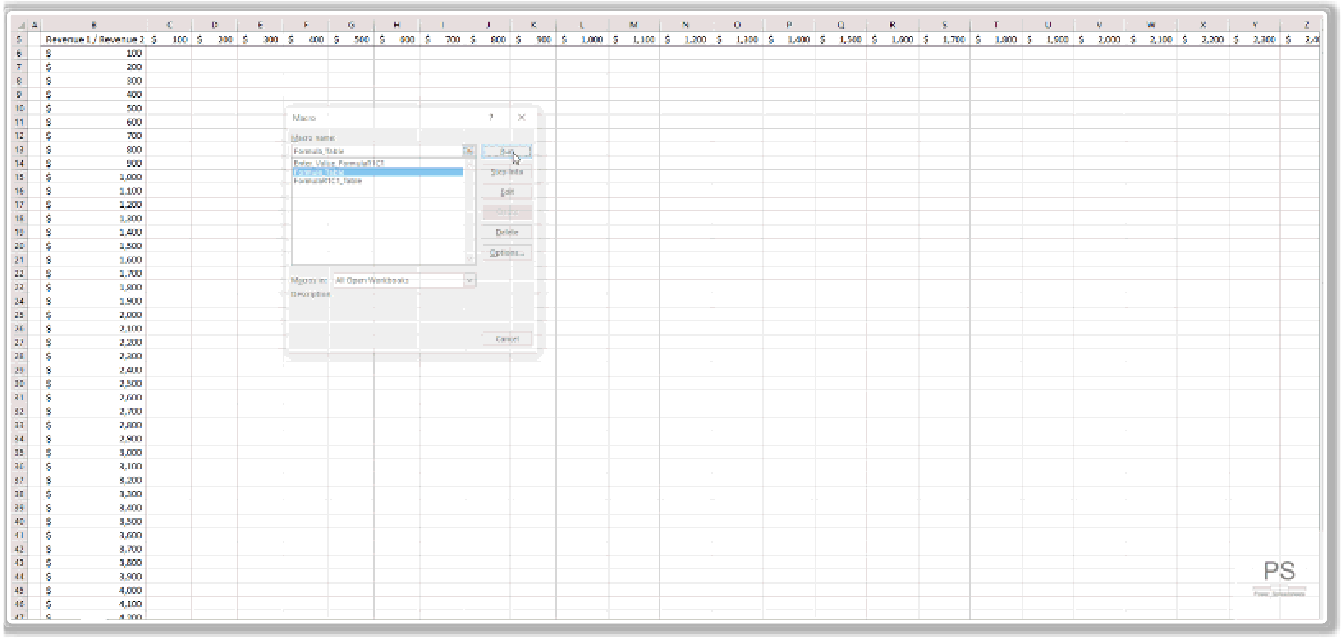


```
Sub Formula_Table()

    Range("C6:CX105").Formula = "=SUM(C$5,$B6)"

End Sub
```

- **Difference #1:** FormulaR1C1_Table uses the Range.FormulaR1C1 property. Formula_Table uses the Range.Formula property.
- **Difference #2:** As a consequence of the above, the formula set by FormulaR1C1_Table uses R1C1-style notation. The formula set by Formula_Table uses A1-style notation.

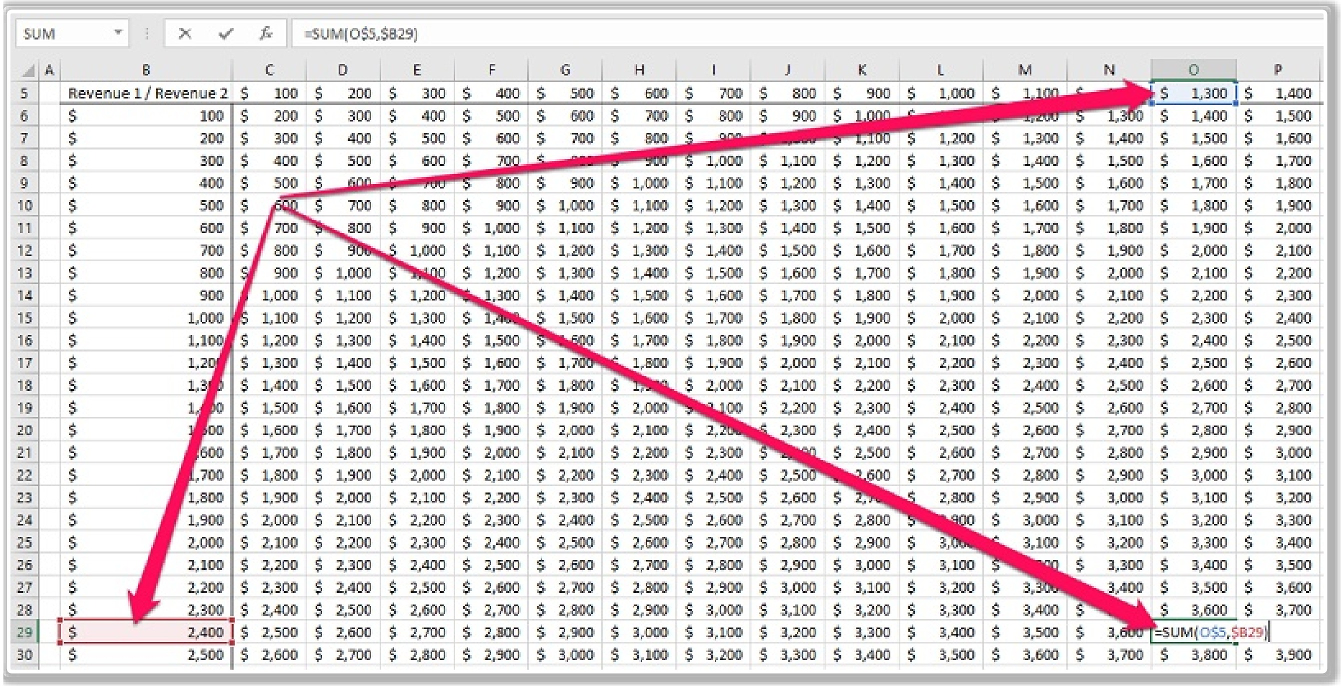
first cell of the table (cell C6). As I explain further below, **Excel automatically adjusts the formula for the other cells** in the table.

The following GIF shows the what happens in Excel when I execute the sample Formula_Table macro. Notice that the results are substantially the same as those obtained when executing the FormulaR1C1_Table macro above.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
5		Revenue 1 / Revenue 2	$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,4
6		$ 100																								
7		$ 200																								
8		$ 300																								
9		$ 400																								
10		$ 500																								
11		$ 600																								
12		$ 700																								
13		$ 800																								
14		$ 900																								
15		$ 1,000																								
16		$ 1,100																								
17		$ 1,200																								
18		$ 1,300																								
19		$ 1,400																								
20		$ 1,500																								
21		$ 1,600																								
22		$ 1,700																								
23		$ 1,800																								
24		$ 1,900																								
25		$ 2,000																								
26		$ 2,100																								
27		$ 2,200																								
28		$ 2,300																								
29		$ 2,400																								
30		$ 2,500																								
31		$ 2,600																								
32		$ 2,700																								
33		$ 2,800																								
34		$ 2,900																								
35		$ 3,000																								
36		$ 3,100																								
37		$ 3,200																								
38		$ 3,300																								
39		$ 3,400																								
40		$ 3,500																								
41		$ 3,600																								
42		$ 3,700																								
43		$ 3,800																								
44		$ 3,900																								
45		$ 4,000																								
46		$ 4,100																								
47		$ 4,200																								

You can also check the formula in cell O29 again (as I did above). The formula is exactly the same as that obtained when carrying out the process manually or executing the FormulaR1C1_Table macro (=SUM(O$5,$B29)).



=SUM(O$5,$B29)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
5		Revenue 1 / Revenue 2	$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ [illegible]	$ 1,300	$ 1,400
6		$ 100	$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	[illegible]	1,200	$ 1,300	$ 1,400	$ 1,500
7		$ 200	$ 300	$ 400	$ 500	$ 600	$ 700	$ 800	$ 900	1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600
8		$ 300	$ 400	$ 500	$ 600	$ 700	[illegible]	900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700
9		$ 400	$ 500	$ 600	700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800
10		$ 500	$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900
11		$ 600	$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000
12		$ 700	$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100
13		$ 800	$ 900	$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200
14		$ 900	1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300
15		$ 1,000	$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400
16		$ 1,100	$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500
17		$ 1,200	$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600
18		$ 1,300	$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700
19		$ 1,400	$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800
20		$ 1,500	$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900
21		$ 1,600	$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000
22		$ 1,700	$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100
23		$ 1,800	$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200
24		$ 1,900	$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300
25		$ 2,000	$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400
26		$ 2,100	$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ [illegible]	$ 3,300	$ 3,400	$ 3,500
27		$ 2,200	$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	3,400	$ 3,500	$ 3,600
28		$ 2,300	$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ [illegible]	$ 3,600	$ 3,700
29		$ 2,400	$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	=SUM(O$5,$B29)	
30		$ 2,500	$ 2,600	$ 2,700	$ 2,800	$ 2,900	$ 3,000	$ 3,100	$ 3,200	$ 3,300	$ 3,400	$ 3,500	$ 3,600	$ 3,700	$ 3,800	$ 3,900

- **Step #1:** Converts the A1-style formula to an R1C1-style formula, as I explain above.

- **Step #2:** Enters the converted R1C1-style formula in the entire range.

- **Step #3:** Displays the R1C1-style formulas using A1-style notation.

The consequence of this is that the macro **roughly mirrors the manual process** that I explain above. That 2-step process is as follows:

- **Step #1:** The relevant formula is entered in the first cell of the table (cell C6).

- **Step #2:** The formula is copied and pasted in all the cells of the table. Excel automatically adjusts the cell references.

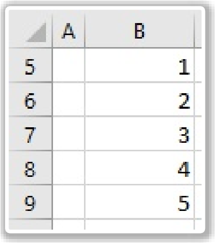
So, strictly speaking, **you can use both the FormulaR1C1 and the Formula properties** to enter a formula in a particular range of cells.

Macros That Rely On Relative References: Avoid This Error

In a previous section, I provide a thorough explanation of how to use relative references when working with the R1C1-style notation.

When working with macros for purposes of creating relative references, there's a particular **behavior that you need to be aware of as it can be the source of mistakes**. This is more thoroughly explained by experts Bill Jelen (Mr. Excel) and Tracy Syrstad in *Excel 2016 VBA and Macros*.

To understand how this can happen, let's go back to the example I provide at the beginning of this blog post. At that point, I enter numbers 1 through 5 in cells B5 to B9.



	A	B
5		1
6		2
7		3
8		4
9		5

For the sake of this example, I create the following **sample macro** (FormulaR1C1_Error).

```
End Sub
```

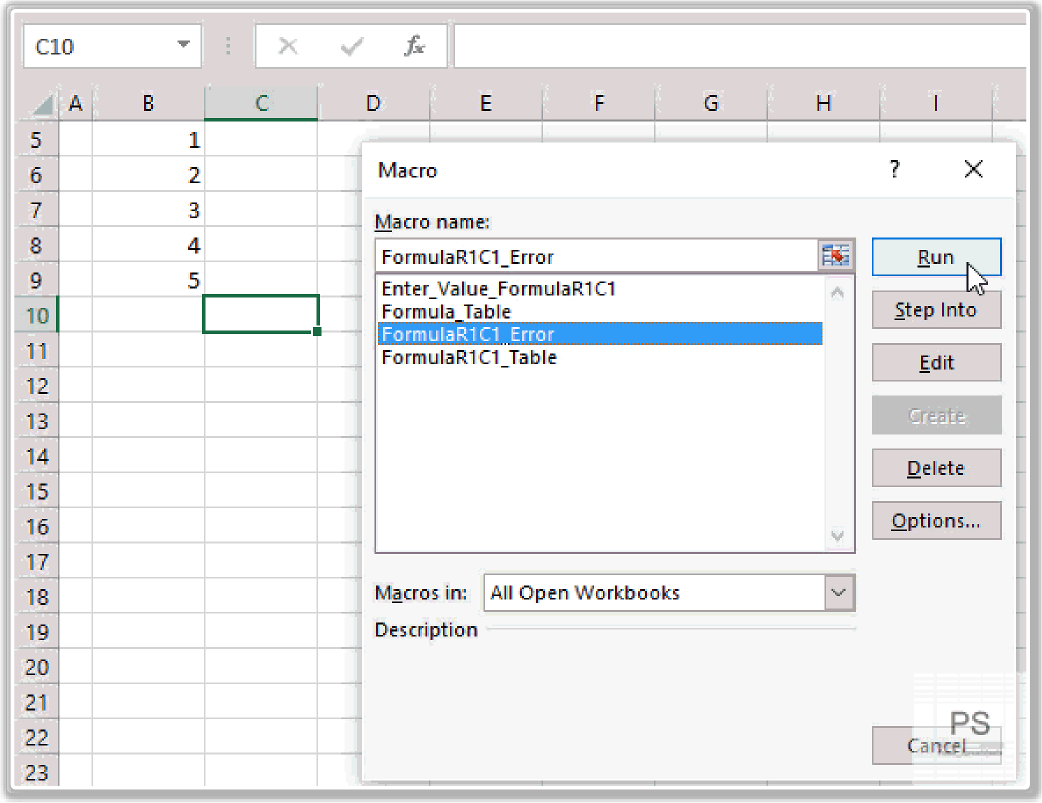
This sample macro proceeds as follows:

- **Step #1:** Uses the Application.ActiveCell property to return the current active cell.

- **Step #2:** Uses the Range.FormulaR1C1 property to set the formula of the active cell obtained in step #1 above. The formula is, simply, a relative reference to the cell located 5 rows above (R[-5]) and 1 column to the left (C[-1]) of the current active cell.

This macro isn't extremely useful for practical purposes. I'm just using it to get a point across

The following GIF shows what happens when I execute the macro while the active cell is cell C10. Notice that, as expected, Excel enters a reference to the cell located 5 rows above and 1 column to the left of C10. That is cell B5, whose value is 1.



Name box: C10

	A	B	C
5		1	
6		2	
7		3	
8		4	
9		5	
10			

Macro name: FormulaR1C1_Error

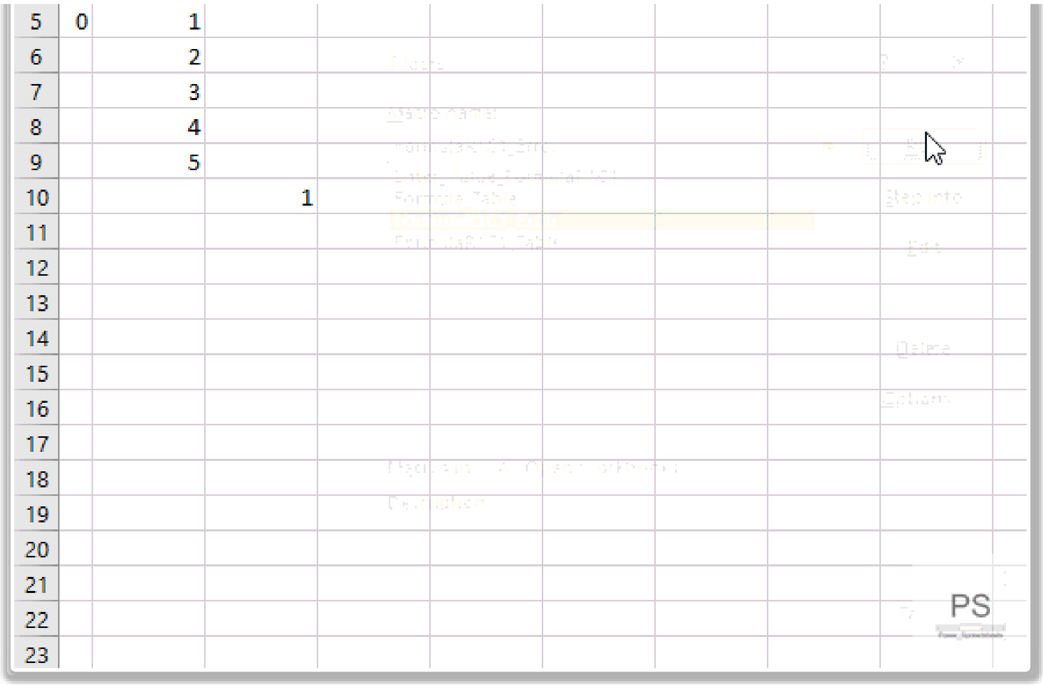
Enter_Value_FormulaR1C1
Formula_Table
FormulaR1C1_Error
FormulaR1C1_Table

Macros in: All Open Workbooks

Description

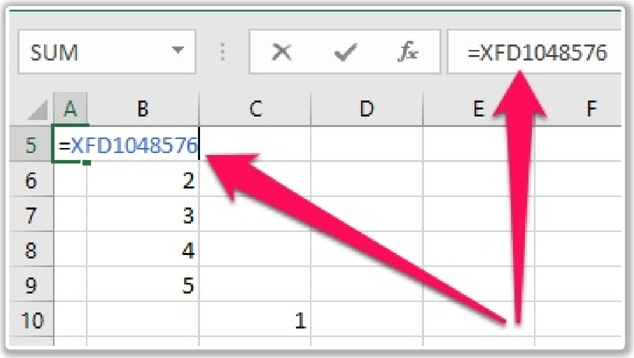
The macro works properly. If you execute the macro for the 4 cells below cell C10 (C11 to C14), the macro enters references to the values that are already entered in cells B6 to B9.

However, take a look at what happens when I execute this macro while the active cell is A5:



5	0	1	
6		2	
7		3	
8		4	
9		5	
10			1
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			

Notice that **Excel doesn't return an error**. It rather **makes reference to cell XFD1048576**.



=XFD1048576

	A	B	C	D	E	F
5	=XFD1048576					
6		2				
7		3				
8		4				
9		5				
10			1			

What's the bottom line?

In such situation, Excel goes around the worksheet. Therefore:

- If your macro makes **reference to a cell that is "above" row 1, Excel wraps around and goes to** the last row of the worksheet to continue searching for the referred cell.
- If your macro makes **reference to a cell that is to "the left" of column A (or column 1 in R1C1-style notation), Excel wraps around and goes to** the last column of the worksheet to continue searching for the relevant cell.

In the example above, both things of these things happened:

- The reference was to a cell 1 column to "the left" of column A (or 1 in R1C1-style). When Excel wraps around, this results in the last column of the worksheet (XFD in A1-style or 16,384 in R1C1-style notation).

The consequence of all of this is that Excel creates the reference that appears in the screenshot above to cell XFD1048576.

I assume that, unless you're facing a very particular situation or scenario, this isn't the behavior you want from your macros.

Conclusion

If you've completed this VBA tutorial, **you now have enough knowledge to start using the R1C1-style notation and the Range.FormulaR1C1 property for purposes of making your Excel macros more efficient, flexible and powerful**. Among other topics, **you know**:

- What are the A1 and the R1C1-style notations.

- Why are the R1C1-style notation and the FormulaR1C1 property useful and important.

- How you can create absolute, relative and mixed references when working with the R1C1-style, and a particular behavior that you should take into consideration when creating relative references within VBA to avoid potential mistakes.

- What are the main characteristics of the Formula.R1C1 property.

- What is the Range.FormulaR1C1Local property, and how you can use it to deal with particular language considerations that arise if you work in a multilingual environment.

In addition to the above, you've **seen a practical example showing how you can use the Range.FormulaR1C1** (or alternatively the Range.Formula) property for purposes of specifying the formula of a large range of cells. Remember that you can **get immediate free access to the Excel workbook that contains the data and macros used throughout this VBA tutorial (including the example) by** clicking here.

Now, before you leave, **please make sure to take a few seconds to leave a comment below** sharing whether…

As I mention above, the Range.FormulaR1C1 property has the potential to help you create more efficient and flexible VBA code. This doesn't mean that you must always rely on the FormulaR1C1 property when working with Visual Basic for Applications. In fact, as I show in an early section of this blog post, some Excel experts and power users don't use the R1C1-style all the time.

Considering this, I would be quite interested in reading your opinion. In particular, I would love to know:

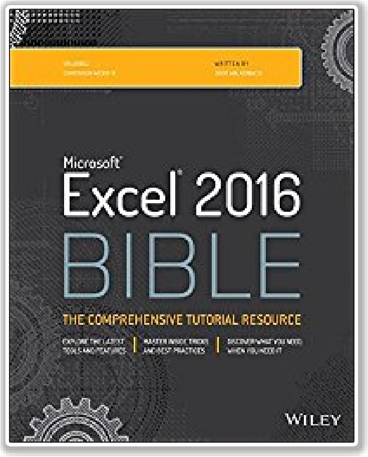
- *In which scenarios do you use the FormulaR1C1 property? In which cases do you prefer relying on the Formula property?*

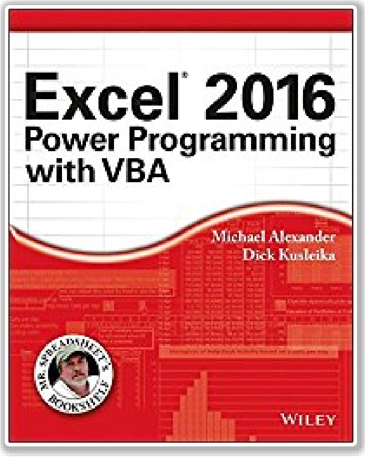
- If you use particular criteria for choosing between the FormulaR1C1 and the Formula property, *what are these criteria you use?*

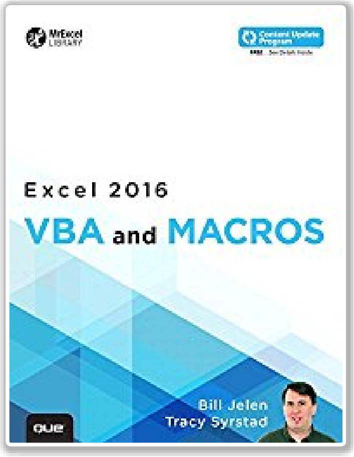
So make sure to **leave a comment below (it only takes you a few seconds) now**. Me and the readers of Power Spreadsheets appreciate the opportunity to learn from you. **Thanks in advance!**

Books Referenced In This Excel Tutorial

Click on any of the images below to purchase the book at Amazon.







Comments for this thread are now closed. ×

12 Comments **Power Spreadsheets** 1 **Login**

♡ **Recommend** 1 **Share** **Sort by Newest**

This was EXTREMELY helpful and just what I was looking for to understand the row and column notation in VBA. Thank you for taking the time to write and post this!!! I have been using excel for a long time, though haven't used VBA in a few years and need to refresh my memory. I had no trouble following and it made sense to me. I plan to read more of your blogs.

Again, many thanks!
Karen

⌃ | ⌄ • Share ›

Jorge A. Gomez Mod ➦ Karen Hill • 8 months ago

Hi Karen,

Thank you very much for your kind words! I'm happy to read that you've found the post helpful.

I think it's great that you're refreshing your memory :-) If at any time I can be of further help as you move along this process, please let me know.

Best,

Jorge

⌃ | ⌄ • Share ›

Alan Elston • a year ago

Hi Jorge,

My Email address is registered, but I do not see how I can get hold of the workbook accompanying this blog. When I click on ...” clicking here “..... I am just asked to register and then told that I already am. I cannot see any directions on how to get hold of any workbook

Alan

⌃ | ⌄ • Share ›

Jorge A. Gomez Mod ➦ Alan Elston • a year ago

Hi Alan,

Thanks for your message. Please accept my apologies for the inconvenience accessing the workbooks.

I just saw your email and sent the workbooks back.

More generally, you can currently access the files through a link that I provide in all the emails that I send (towards the end, as a PS). The link is also included in the final welcome email (if you still have it).

If you experience any further issues, please feel free to let me know by email. I'm currently working on changing a few things, including the way in which the workbooks are accessed. Hopefully, once this is up and running, things will be easier for you and other readers.

Thanks for the support and for being part of our Newsletter.

Jorge

⌃ | ⌄ • Share ›

A couple of things confused me as a beginner, and in helping others to understand I have found it worth noting:

_1) The expression R1C1 is used generally to refer to the R C type convention or “Style”, both the Absolute and Relative versions. R1C1 in a formula usage is actually an Absolute version.

The .FormulaR1C1 Property, for example can be used, for example, to assign a Relative R C type such as R[1]C[1] and also an Absolute type such as R1C2, (or mixed)

_2) It is good to note that something like R[1]C is a “shorthand” for R[1]C[0]. Both give the same result.

Alan Elston

⌃ | ⌄ • Share ›

Jorge A. Gomez Mod ➦ Alan Elston • a year ago

Thank you very much for your nice comment and suggestions Alan :-).

I agree that both tips you provide are very useful clarifications. I'm planning to make some amendments and clarifications to these posts in the future. I will include and highlight your ideas in the article itself (with credit to you, obviously).

Thanks again for stopping by!

1 ⌃ | ⌄ • Share ›

Alan Elston ➦ Jorge A. Gomez • a year ago

You are very welcome, thanks for the feedback :)
Alan

⌃ | ⌄ • Share ›

PS

This comment is awaiting moderation. Show comment.

Jorge A. Gomez Mod ➦ Amit Kumar • 6 months ago

Many thanks for your comment, Amit! I'm very happy to read you've found the article helpful.

⌃ | ⌄ • Share ›

Mourad Louha • a year ago

Hi Jorge,
awesome tutorial - and thank you too for linking and recommending the Excel-Translator.
I will link back to here :-)

Best Regards,
Mourad Louha

⌃ | ⌄ • Share ›

Jorge A. Gomez Mod ➦ Mourad Louha • a year ago

Hi Mourad,

Thank you very much for stopping by and your kind comments. I'm very happy to read these nice comments from you!

I'm also happy to link/recommend the Excel-Translator whenever I get the chance. It's a great

Jorge

˄ | ˅ • Share ›

Janmartens ➦ Jorge A. Gomez • a year ago

Hi Jorge, I have this practical Tip from Ozgrid :
To get an RC formula, type the A1 formula in in any cell then select the cell, go Tools>Macro>Record new macro and record a macro pushing F2 then Enter.
Optimize Slow VBA Code. Speed Up Efficient VBA Code/Macros

˄ | ˅ • Share ›

Jorge A. Gomez Mod ➦ Janmartens • a year ago

Hi Jan,

I agree that's quite a useful tip. Particularly, since most of us don't use R1C1 notation outside VBA, sometimes building the formulas can feel a little un-intuitive. This tip helps a lot to get the formula without actually having to write it in R1C1-style from scratch.

Thanks a lot for sharing!

˄ | ˅ • Share ›

✉ **Subscribe** **Add Disqus to your siteAdd DisqusAdd** **Privacy**

POWER SPREADSHEETS IN SOCIAL MEDIA

Excel Resources | Excel Shortcuts

Contact

Privacy Policy | Terms and Conditions | Limit of Liability and Disclaimer of Warranty

Excel ® is a registered trademark of the Microsoft Corporation. Power Spreadsheets is not affiliated with the Microsoft Corporation. Some of the links in Power Spreadsheets are affiliate links. If you make a purchase through one of them, I may receive a commission. This commission comes at no additional cost to you,