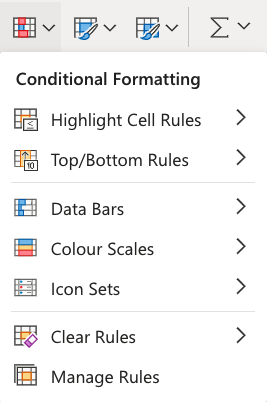
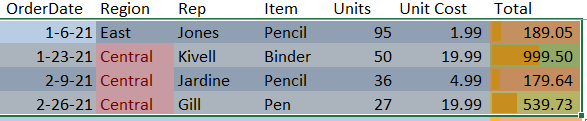
1. CONDITIONAL FORMATTING, IF, COUNTIF, SUMIF, AVERAGE,

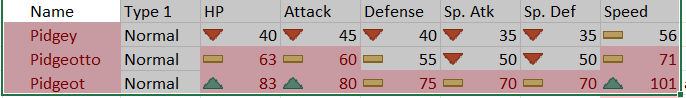
CONCAT using excel formulas

1. Conditional formatting rule:

Conditional formatting is used to change the appearance of cells in a range based on your specified **conditions**.







IF Function

The **IF** function is a premade function in Excel, which returns values based on a **true** or **false** **condition**.

It is typed =IF and has 3 parts:

=IF(**logical\_test**, **[value\_if\_true]**, **[value\_if\_false]**)

Example: =IF(E3>50,"Yes", "No")

## COUNTIF Function

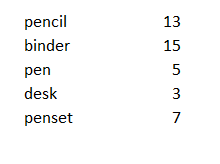
The **COUNTIF** function is a premade function in Excel, which counts cells as specified.

It is typed =COUNTIF

How to use the =COUNTIF function:

1. Select a cell
2. Type =COUNTIF
3. Double click the **COUNTIF** command
4. Select a range
5. Type ,
6. Select a cell (the criteria, the value that you want to count)
7. Hit enter

Example: =COUNTIF(Table\_1[Item],D2)



## SUMIF Function

The **SUMIF** function is a premade function in Excel, which calculates the sum of values in a range based on a **true** or **false** **condition**.

It is typed =SUMIF:

Syntax:

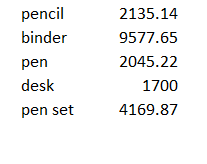
=SUMIF(**range**, **criteria**, **[sum\_range]**)

The **condition** is referred to as criteria, which can check things like:

* If a number is **greater than** another number >
* If a number is **smaller than** another number <
* If a number or text is **equal** to something =

The [sum\_range] is the range where the function calculates the sum.

Example: =SUMIF(Table\_1[Item],J13,Table\_1[Total])



## AVERAGE Function

The **AVERAGE** function is a premade function in Excel, which calculates the average (arithmetic mean).

It is typed =AVERAGE

It adds the range and divides it by the number of observations.

**Example:**

The average of (2, 3, 4) is 3.  
3 observations (2, 3 and 4)  
The sum of the observations (2 + 3 + 4 = 9)  
(9 / 3 = 3)  
The average is 3

How to use the =AVERAGE function, step by step:

1. Select a cell (F2)
2. Type =AVERAGE
3. Double click the **AVERAGE** command
4. Select a range (B2:E2)
5. Hit enter
6. Next, Fill F2:F7

Example: =AVERAGE(D3:I3)

## CONCAT Function

To concatenate is to link something together.

CONCAT is a function in Excel and is short for concatenate.

The **CONCAT** function is used to link multiple cells without adding any delimiters between the combined cell values.

It is typed =CONCAT

=CONCAT(**cell1**, **delimiter**, **cell2**)

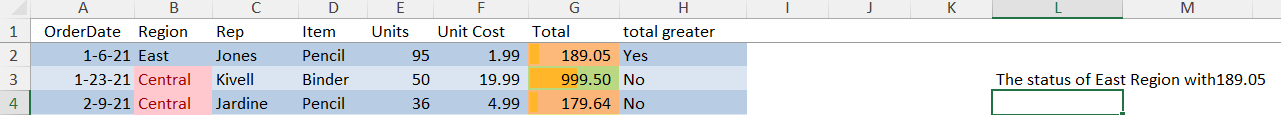
### **Example of delimiters**

comma , semicolon ; quotes " or ' braces {} pipes | slashes / \

##### **Start the CONCAT function**

1. Select a cell E2
2. Type =CONCAT
3. Double click the **CONCAT** command

Example: =CONCAT("The status of ",B2," Region with",G2)



1. INDEX, MATCH, UNIQUE, IFS, COUNTIFS, SUMIFS, AVERAGEIFS using Excel formulas

## IFS Function

The **IFS** function is a premade function in Excel, which returns values based on one or more **true** or **false** **conditions**.

It is typed =IFS and has two or more parts:

Syntax:

=IFS(**logical\_test1**, **value\_if\_true1**, **[logical\_test2, value\_if\_true2]**, **[logical\_test3;** ...)

The **conditions** are referred to as logical\_test1, logical\_test2, ..., which can check things like:

* If a number is **greater than** another number >
* If a number is **smaller than** another number <
* If a number or text is **equal** to something =

Each condition is connected with a return value.

1. Select the cell D2
2. Type =IFS
3. Double click the **IFS** command
4. Specify the first condition C2>90
5. Type ,
6. Specify the value "Fast" for when the first condition is **TRUE**
7. Type ,
8. Specify the second condition C2>50
9. Type ,
10. Specify the value "Normal" for when the second condition is **TRUE**
11. Type ,
12. Specify the third condition C2<=50
13. Type ,
14. Specify the value "Slow" for when the third condition is **TRUE**
15. Hit enter

Example: =IFS(D14>90, "Fast",D14>50,"Normal",D14<=50,"Slow")



COUNTIFS Function

The **COUNTIFS** function is a premade function in Excel, which counts cells in a range based on one or more **true** or **false** **condition**.

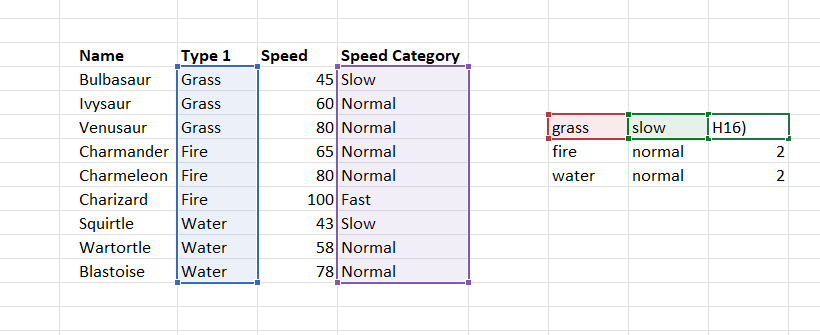
It is typed =COUNTIFS:

=COUNTIFS(**criteria\_range1**, **criteria1**, **[criteria\_range2, criteria2]**, ...)

The **conditions** are referred to as critera1, criteria2, .. and so on, which can check things like:

* If a number is **greater than** another number >
* If a number is **smaller than** another number <
* If a number or text is **equal** to something =

Example: =COUNTIFS(C14:C22,G16,E14:E22,H16)



AVERAGEIFS Function

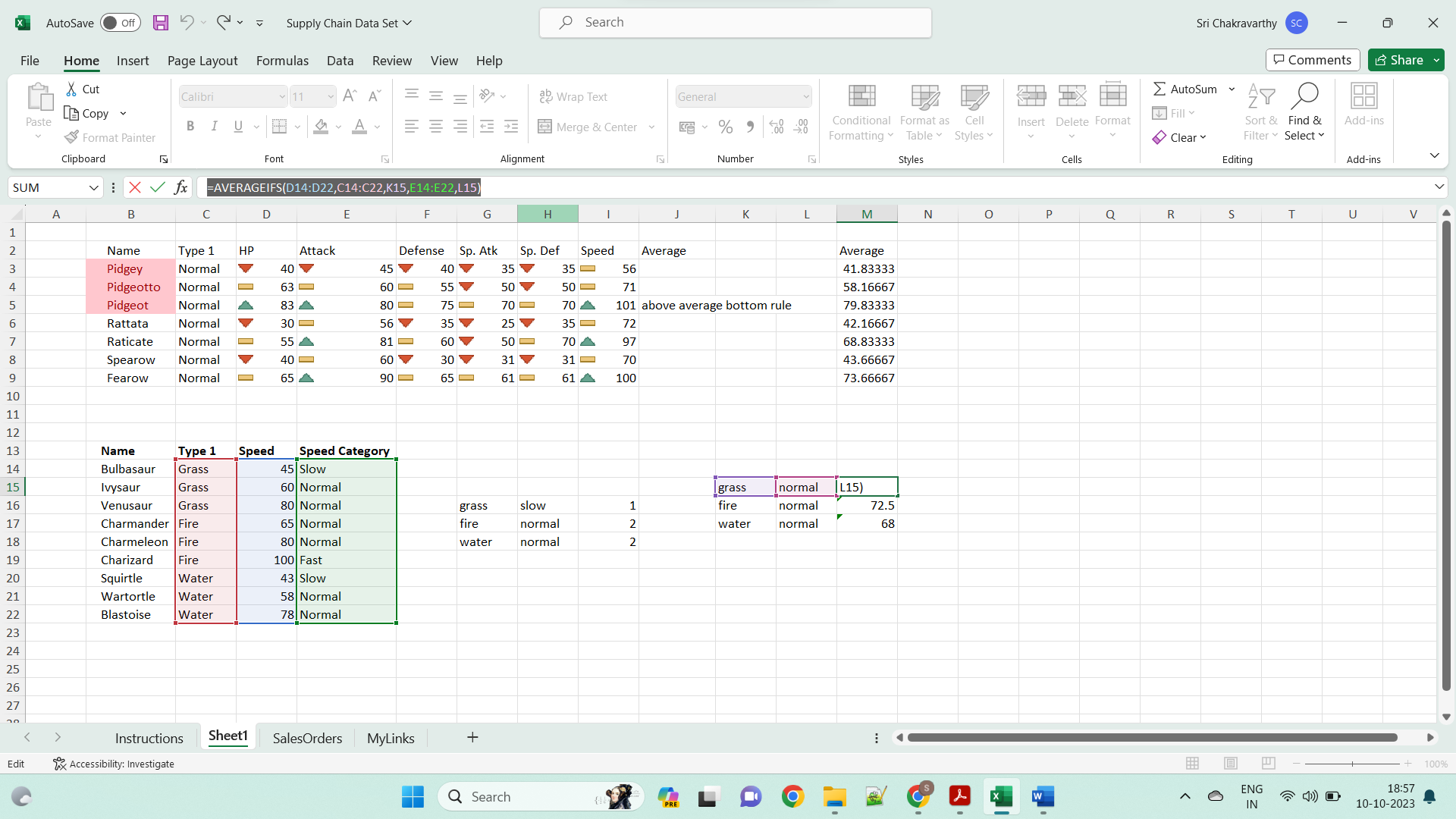
The **AVERAGEIFS** function is a premade function in Excel, which calculates the [average](https://www.w3schools.com/statistics/statistics_mean.php) of a range based on one or more **true** or **false** **condition**.

It is typed =AVERAGEIFS:

=AVERAGEIFS(**average\_range**, **criteria\_range1**, **criteria1, ...**)

The **conditions** are referred to as critera1, criteria2, .. and so on, which can check things like:

* If a number is **greater than** another number >
* If a number is **smaller than** another number <
* If a number or text is **equal** to something =



SUMIFS Function

The **SUMIFS** function is a premade function in Excel, which calculates the sum of a range based on one or more **true** or **false** **condition**.

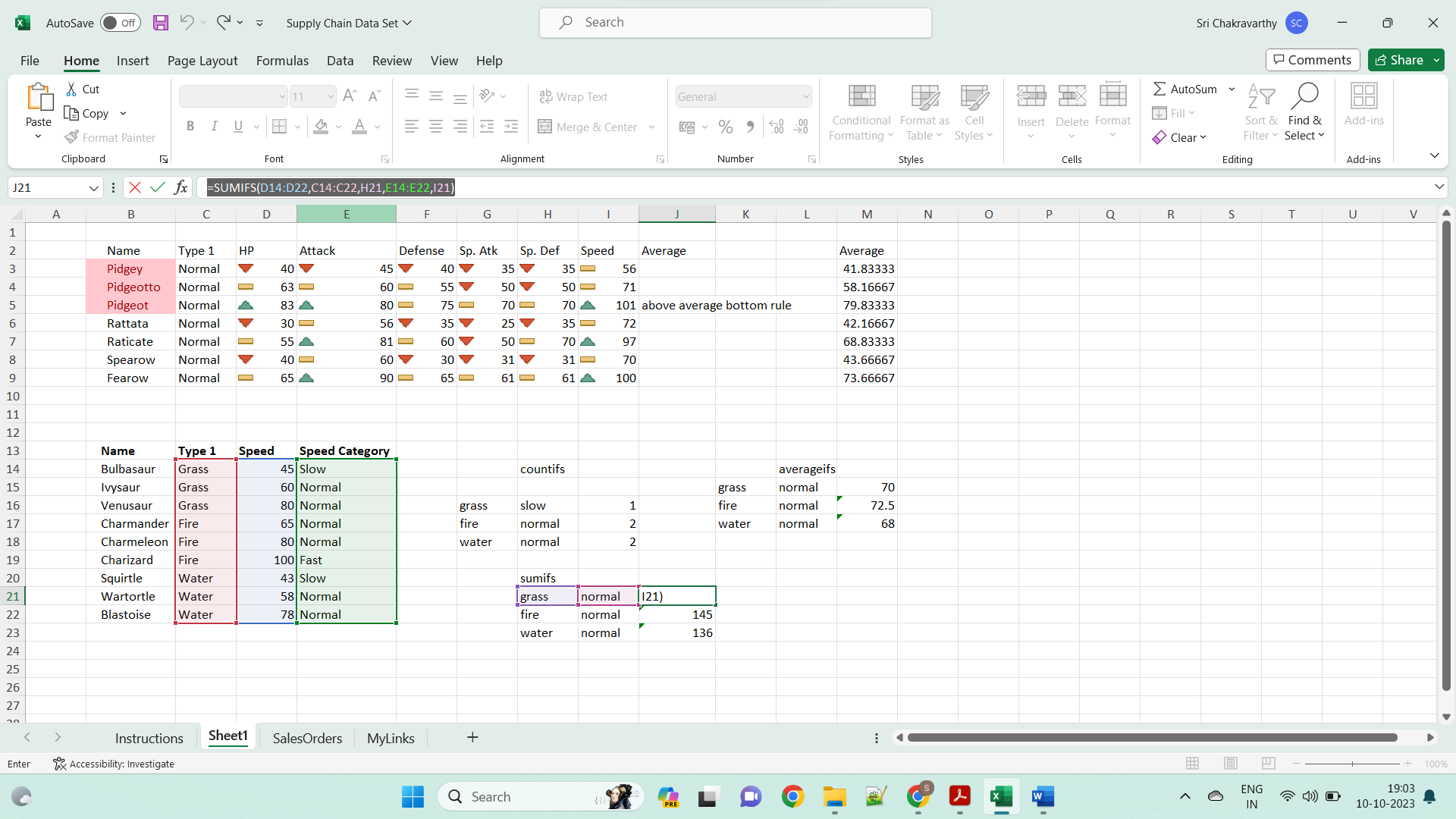
It is typed =SUMIFS:

=SUMIFS(**sum\_range**, **criteria\_range1**, **criteria1**, **[criteria\_range2, criteria2] ...**)

The **conditions** are referred to as criteria1, criteria2, and so on, which can check things like:

* If a number is **greater than** another number >
* If a number is **smaller than** another number <
* If a number or text is **equal** to something =

Example: =SUMIFS(D14:D22,C14:C22,H21,E14:E22,I21)



### **INDEX array form**

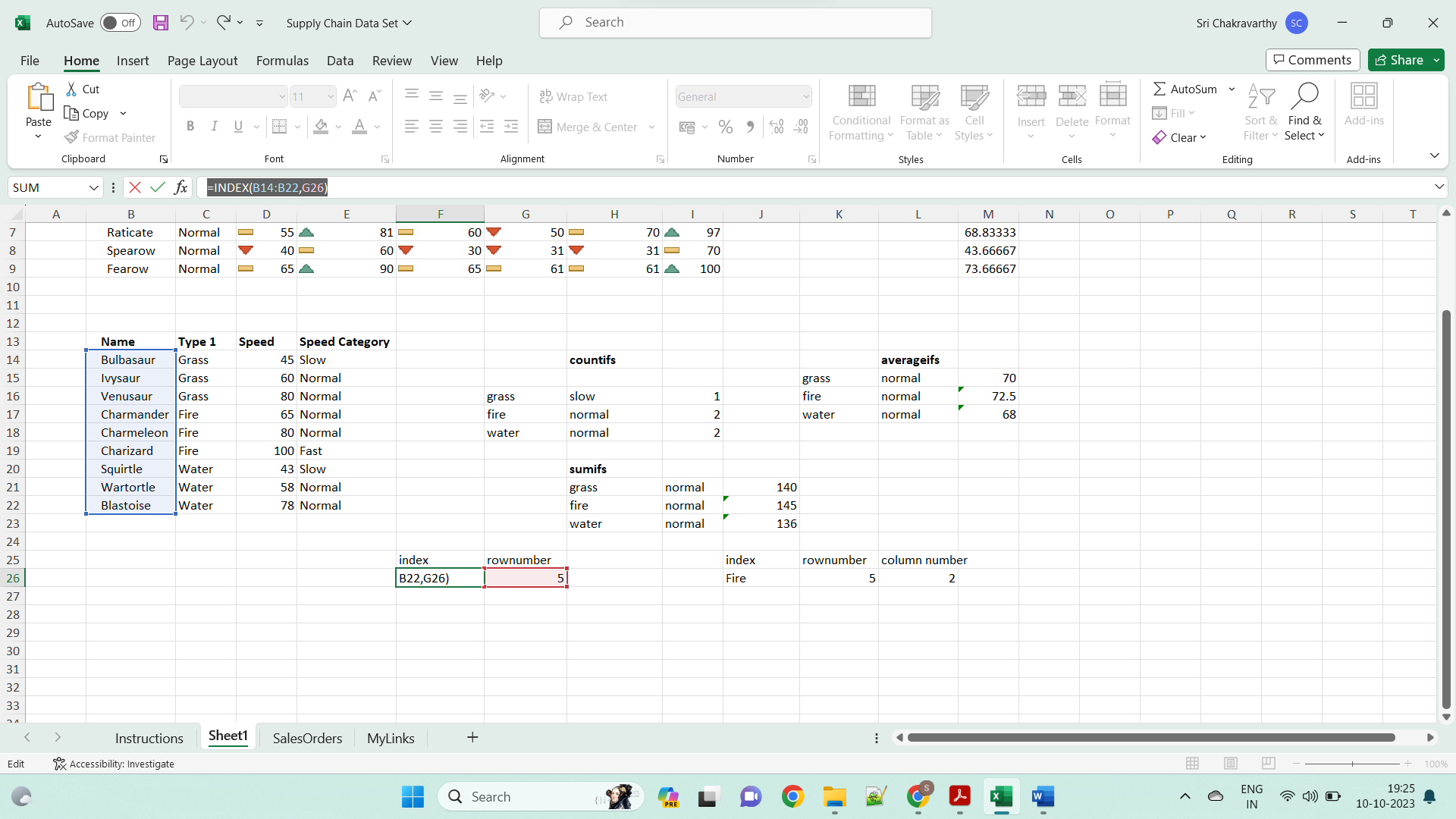
The INDEX array form returns the value of a certain element in a range or array based on the row and column numbers you specify.

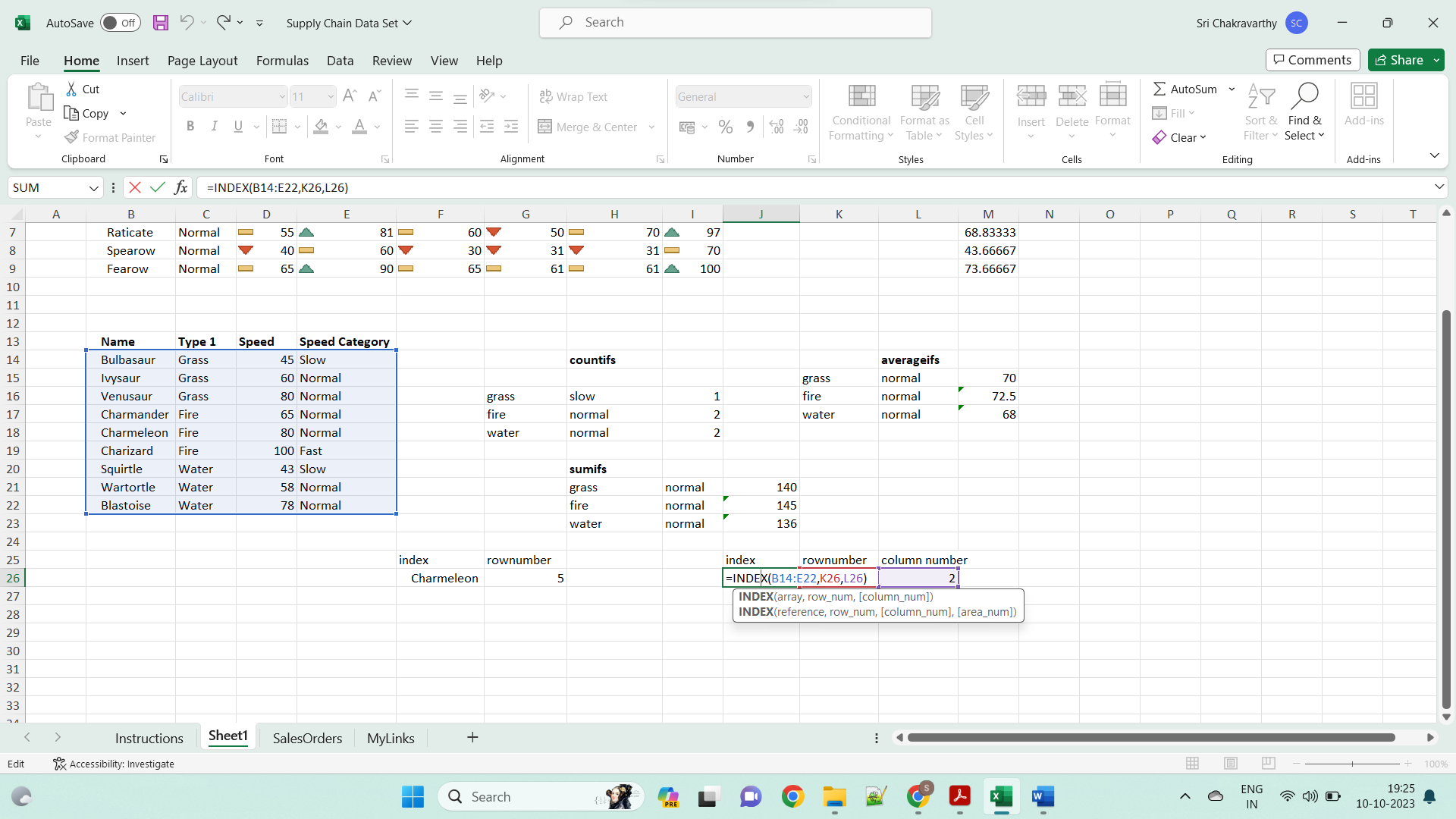
**INDEX(array, row\_num, [column\_num])**

* **array** - is a range of cells, named range, or table.
* **row\_num** - is the row number in the array from which to return a value. If row\_num is omitted, column\_num is required.
* **column\_num** - is the column number from which to return a value. If column\_num is omitted, row\_num is required.

For example, the formula **=INDEX(A1:D6, 4, 3)**

**=INDEX(B14:E22,K26,L26)**





1. Vlookup ,hlookup,xlookup,count,counta

-Find values in a range of data - VLOOKUP and HLOOKUP

-Obtain a value or the reference to a value from within a table or range - INDEX

-Obtain the relative position of a specified item in a range of cells – MATCH

Using VLOOKUP Function

The syntax of the VLOOKUP function is

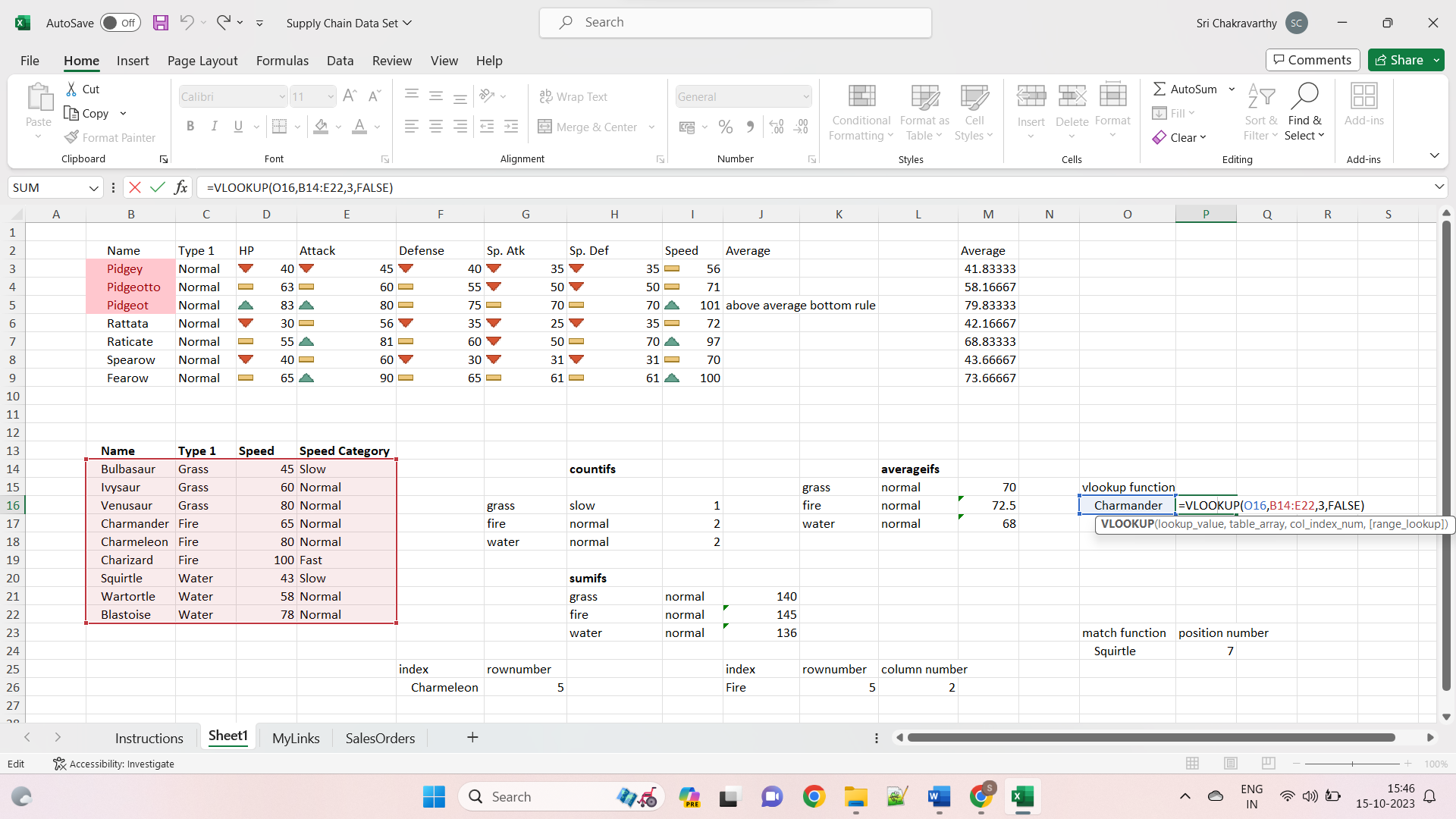
VLOOKUP (lookup\_value, table\_array, col\_index\_num, [range\_lookup])

Where

* **lookup\_value** − is the value you want to look up. Lookup\_value can be a value or a reference to a cell. Lookup\_value must be in the first column of the range of cells you specify in table\_array
* **table\_array** − is the range of cells in which the VLOOKUP will search for the lookup\_value and the return value. table\_array must contain
  + the lookup\_value in the first column, and
  + the return value you want to find

**Note** − The first column containing the lookup\_value can either be sorted in ascending order or not. However, the result will be based on the order of this column.

* **col\_index\_num** − is the column number in the table\_array that contains the return value. The numbers start with 1 for the left-most column of table-array
* **range\_lookup** − is an optional logical value that specifies whether you want VLOOKUP to find an exact match or an approximate match. range\_lookup can be



## Using HLOOKUP Function

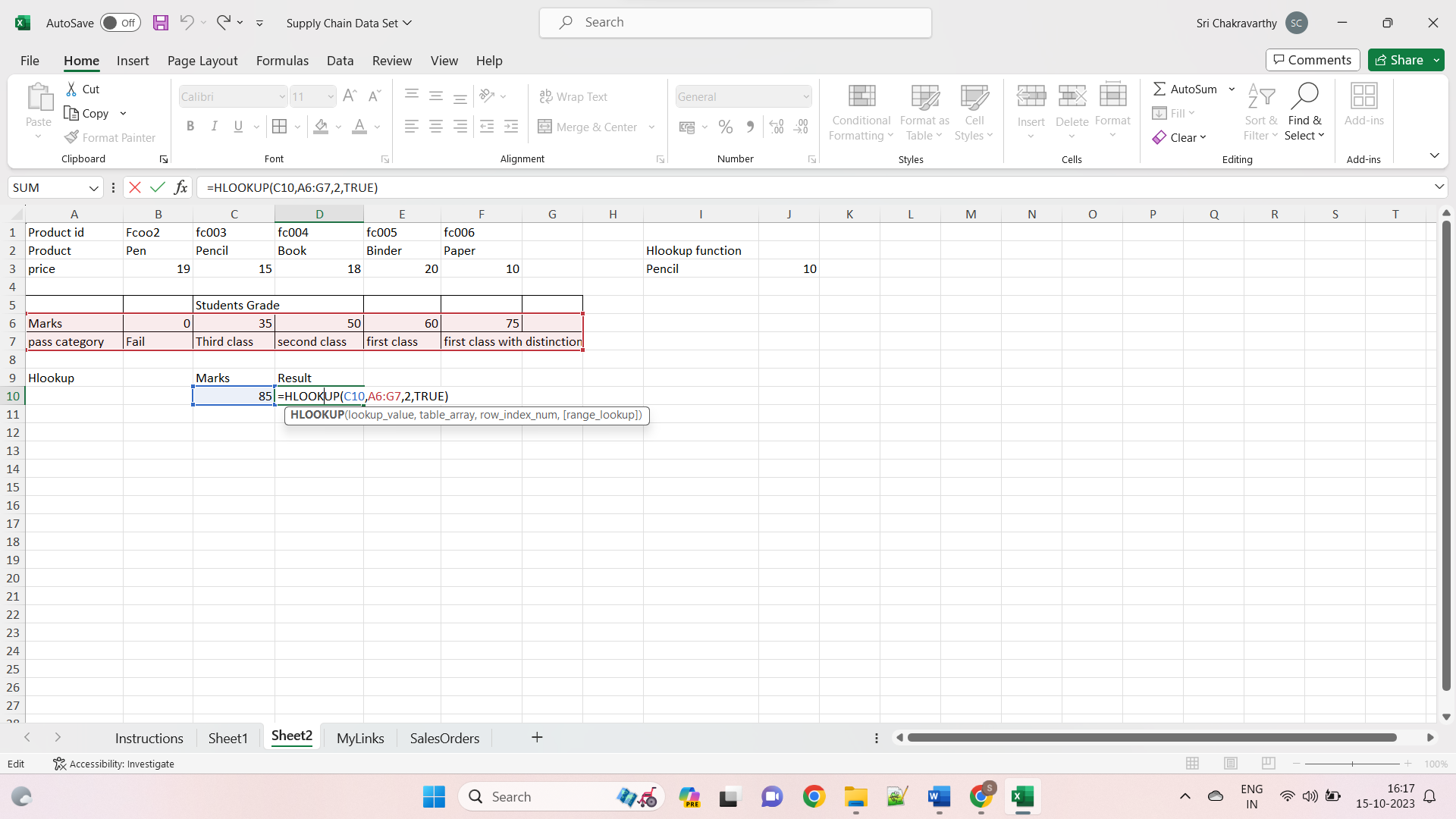
You can use **HLOOKUP** function if the data is in rows rather than columns.

The Syntax of HLOOKUP function is

HLOOKUP (lookup\_value, table\_array, row\_index\_num, [range\_lookup])

Where

* **lookup\_value** − is the value to be found in the first row of the table
* **table\_array** − is a table of information in which data is looked up
* **row\_index\_num** − is the row number in table\_array from which the matching value will be returned
* **range\_lookup** − is a logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match
* **range\_lookup** can be
  + omitted, in which case it is assumed to be TRUE and HLOOKUP tries to find an approximate match
  + TRUE, in which case HLOOKUP tries to find an approximate match. In other words, if an exact match is not found, the next largest value that is less than lookup\_value is returned
  + FALSE, in which case HLOOKUP tries to find an exact match
  + 1, in which case it is assumed to be TRUE and HLOOKUP tries to find an approximate match
  + 0, in which case it is assumed to be FALSE and HLOOKUP tries to find an exact match



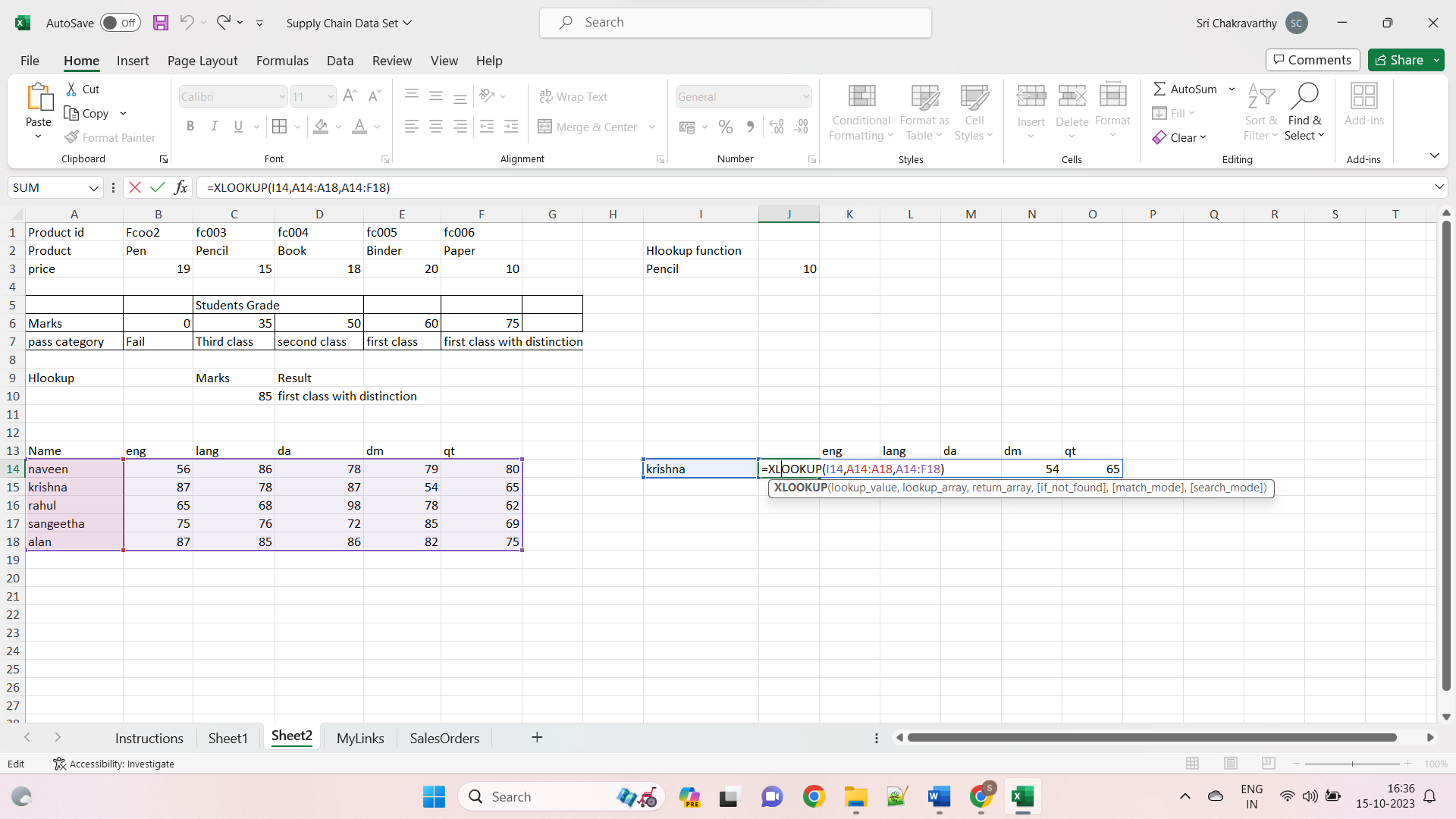
Xlookup function:

The XLOOKUP function is the advanced lookup function that combines the functionalities of both VLOOKUP and HLOOKUP functions. It has no limitations and searches for a value in a row or column of a table. It then returns the corresponding value in a table.

The syntax that applies to the XLOOKUP function is as follows –

**=XLOOKUP(value, lookup\_array, return\_array, [if\_not\_found], [match\_mode], [search\_mode])**

### **Parameters**

* **value**: The entry you are looking for; can be composed of text, numbers, or placeholders. This is a mandatory argument.
* **lookup\_array**: The range or array in which we need to search the required information. Another mandatory argument.
* **return\_array**: The range in which we need to return the desired information. It is also a mandatory argument.
* **if\_not\_found**\*: If Excel doesn’t find a valid match, it can return a text or value that you provide. Otherwise, the program will display the message “#N/A”. Optional argument.
* **match\_mode**\*: Compare mode allows you to specify a type of match. This is an optional argument
* 0: Exact match. If none is found, returns #N/A. This value is the default.
* -1: Exact match. If none is found, it will return the next smallest element.
* 1: Exact match. If none is found, it will return the largest item.
* 2: A wildcard match when \*,? and ~ has a special meaning.
* **search\_mode**\*: This parameter determines where the search should start (“1” for the first element and “-1” for the last) or you start a binary search.
* 1: Performs a search starting with the first element. This value is the default.
* -1: Performs a reverse lookup starting from the last element.
* 2: Performs a binary search that depends on sorting in ascending order. If not sorted, invalid results will be returned.
* -2: Performs a binary search depending on the sort in descending order. If not sorted, invalid results will be returned.
* 

## COUNT Function

The **COUNT** function is a premade function in Excel, which counts cells with numbers in a range.

It is typed =COUNT

**Note:** The **COUNT** function only counts cells with numbers, **not**cells **with letters**. The [COUNTA](https://www.w3schools.com/excel/excel_counta.php) function is better used if the cells have letters.

How to use the =COUNT function:

1. Select a cell
2. Type =COUNT
3. Double click the **COUNT**command
4. Select a range
5. Hit enter



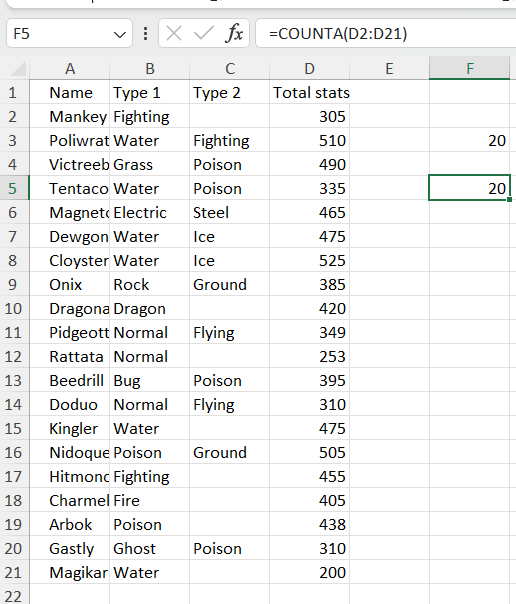
COUNTA Function

The **COUNTA** function is a premade function in Excel, which counts all cells in a range that has values, both numbers and letters.

It is typed =COUNTA

How to use the =COUNTA function:

1. Select a cell
2. Type =COUNTA
3. Double click the **COUNTA** command
4. Select a range



LEFT Function

The **LEFT** function is used to retrieve a chosen amount of characters, counting from the left side of an Excel cell. The chosen number has to be greater than 0 and is set to 1 by default.

It is typed =LEFT

If you want to use the function on a single cell, write:

=LEFT(**cell**)

If you want to use the function on a range of cells, write:

=LEFT(**start cell**:**end cell**)

##### **Start the LEFT function**

1. Select a cell E2
2. Type =LEFT
3. Double click the **LEFT** command



RIGHT Function

The **RIGHT** function is used to retrieve a chosen amount of characters, counting from the right side of an Excel cell. The chosen number has to be greater than 0 and is set to 1 by default.

It is typed =RIGHT

If you want to use the function on a single cell, write:

=RIGHT(**cell**)

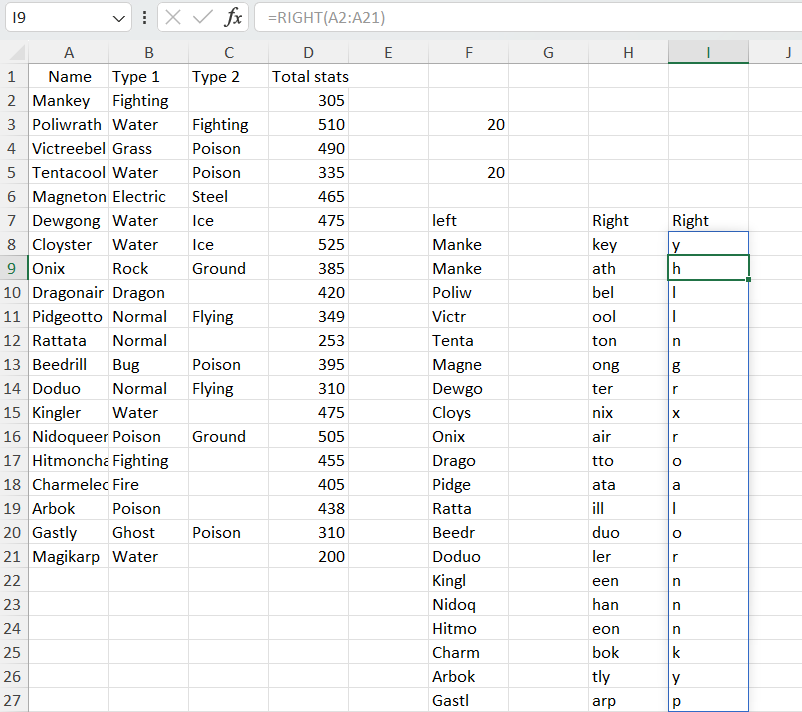
If you want to use the function on a range of cells, write:

=RIGHT(**start cell**:**end cell**)

##### **Start the RIGHT function**

1. Select a cell E2
2. Type =RIGHT
3. Double click the **RIGHT** command





Mid:

The **MID function** returns the specified number of characters in a text string, starting from a specified position (ie. starting from a specified character number). Use this function to extract a sub-string from any part of a text string. ***Syntax: MID(text\_string, start\_number, char\_numbers)***. The **text\_string** argument is the text string from which you want to extract the specified number of characters. The **start\_number** argument specifies the character number from which to start extracting the sub-string, the first character in a text string being start\_number 1 and incrementing towards the right. The **char\_numbers** argument specifies the number of characters to extract from the text string.

Ex: **MID(A1,2,6), MID(A1,2, 15), =MID(A1,0,7)**



Len:

The Excel **LEN function** can be used both as a worksheet function and a VBA function. The **worksheet LEN function** returns the number of characters in a text string. Use this function to get the length of a text string.

***Syntax: LEN(text\_string)***.

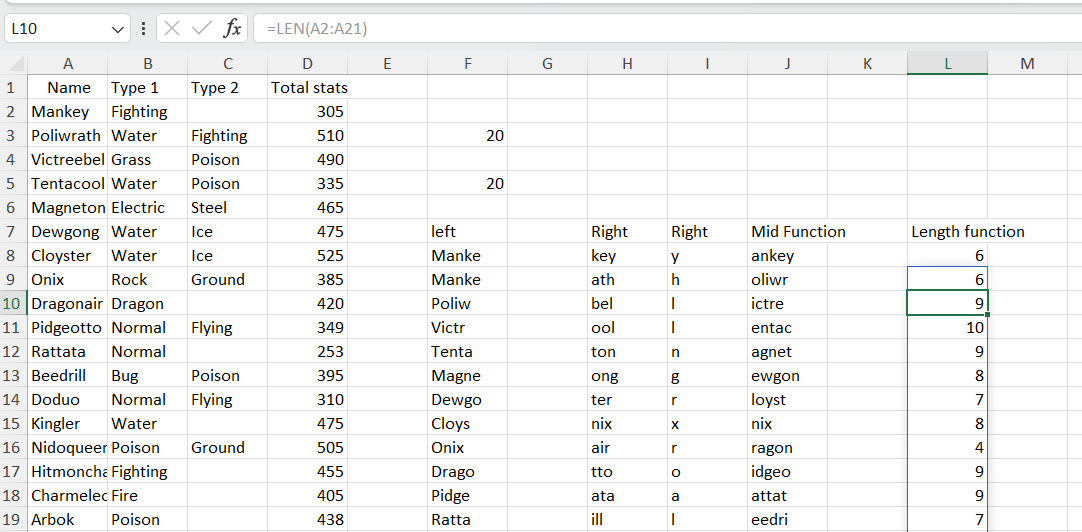
It is necessary to mention the **text\_string** argument which is the text string whose length you want to get in number of characters. Note that spaces also count as characters. The **worksheet LENB function** returns the number of bytes used to represent the characters in a text string.

***Syntax: LENB(text\_string)*** or  ***Syntax: Len(text\_string)*** or ***Len(variable\_name)***

**=LEN(A1).**

**Ex:**

**=SUBSTITUTE( C7,C7,"ice break")**

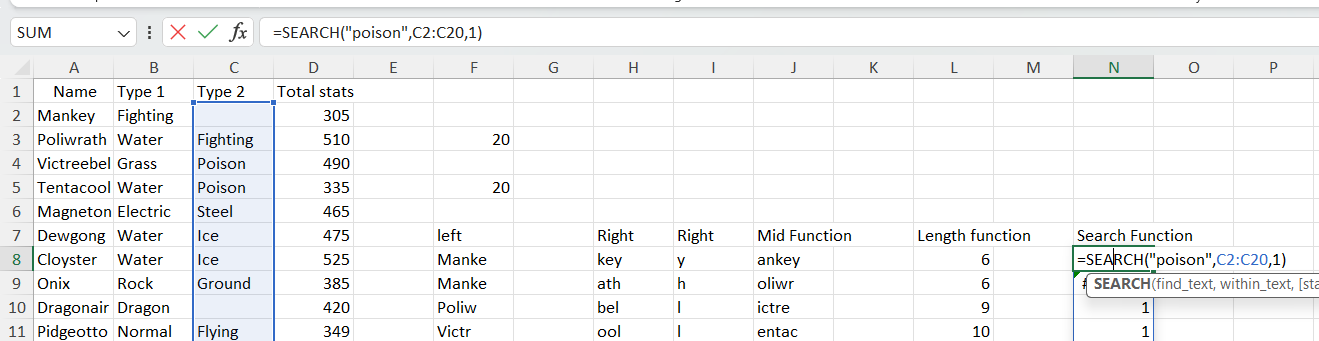


**Search:**

The **SEARCH function** returns the starting position of a text string, which it locates from within a second text string. The starting position is the number, which is calculated beginning from the first character of that second text string. SEARCH is not case-sensitive. ***Syntax: SEARCH(text\_to\_find, within\_text, start\_number)***.

**Ex:=SEARCH("a",A2)**

**=SEARCH("d?y",A5)   
=SEARCH("d\*y", A5)**



Substitute:

Use the **SUBSTITUTE function** to replace old text with new text in a text string. **Substitute Function vs Replace Function**: The difference is that in Substitute you replace specific text with new text while in Replace you replace any text based on its position and length within a text string.

***Syntax: SUBSTITUTE(text, old\_text, new\_text, instance\_num)***.

It is necessary to specify all arguments except instance\_num. The **text** argument is the text string which contains the old text which you want to replace. The **old\_text** argument is the text string which you want to replace with new text. The **new\_text** argument is the new text which replaces the old text. The **instance\_num** argument is a numerical value which specifies the occurrence of old text which you want to replace with new text.

Ex :=SUBSTITUTE( C7,C7,"ice break")

**IS NUMBER:**

* The ISNUMBER function is an information function used to find if the cell value in reference is a numerical value or not. It returns values as “true” or “false.”
* The formula for the ISNUMBER function is “=ISNUMBER (value).”

The ISNUMBER formula returns the following logical values:

* “True,” if the “value” parameter is numeric.
* “False,” if the “value” parameter is non-numeric.

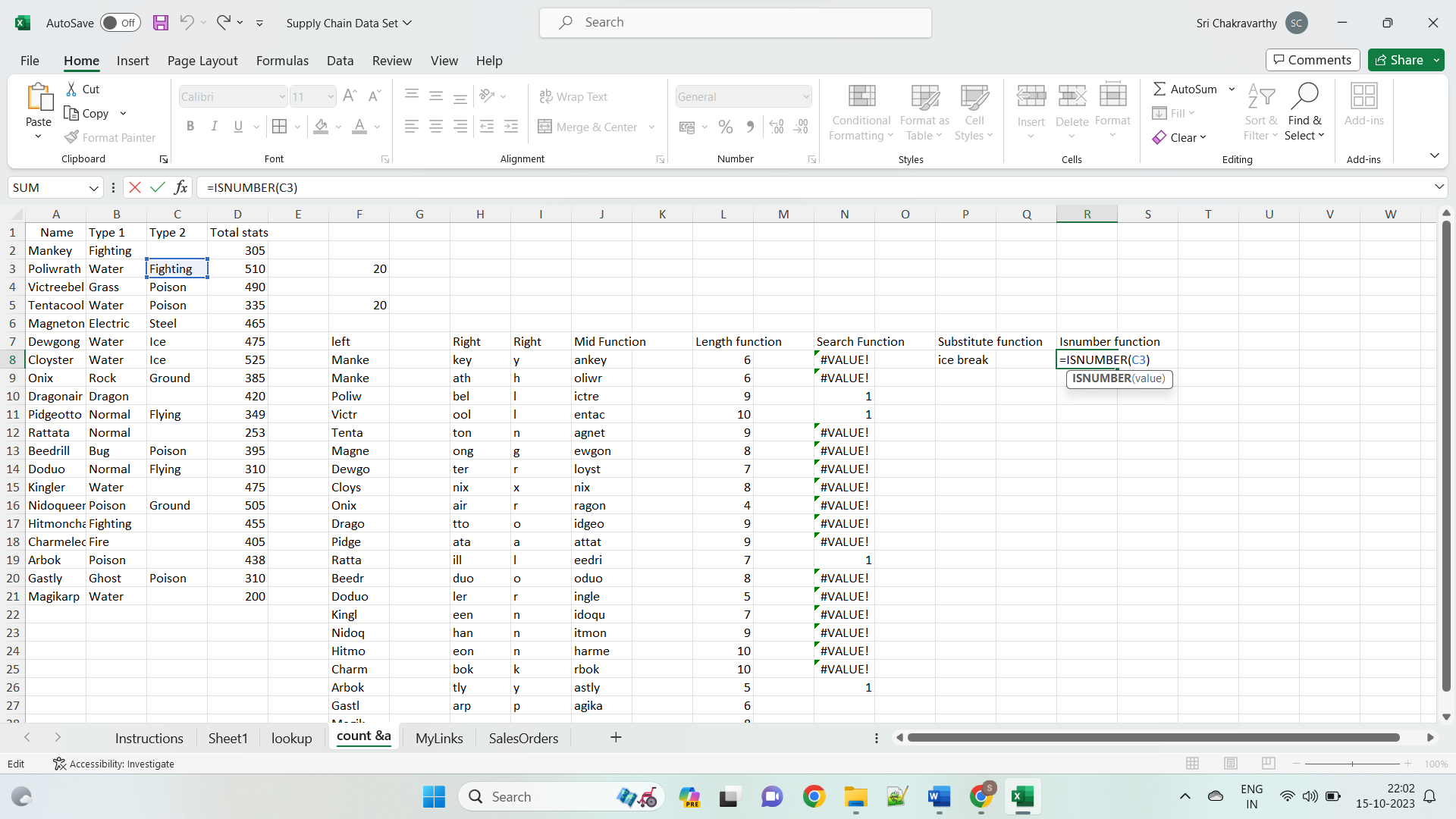
The formula is represented in the following format.

“=ISNUMBER(Reference Cell)”

The argument “reference cell” refers to the cell which we want to check or identify.

Let us consider the following ISNUMBER formula.

“=ISNUMBER(T1XT)”



**Search function:**

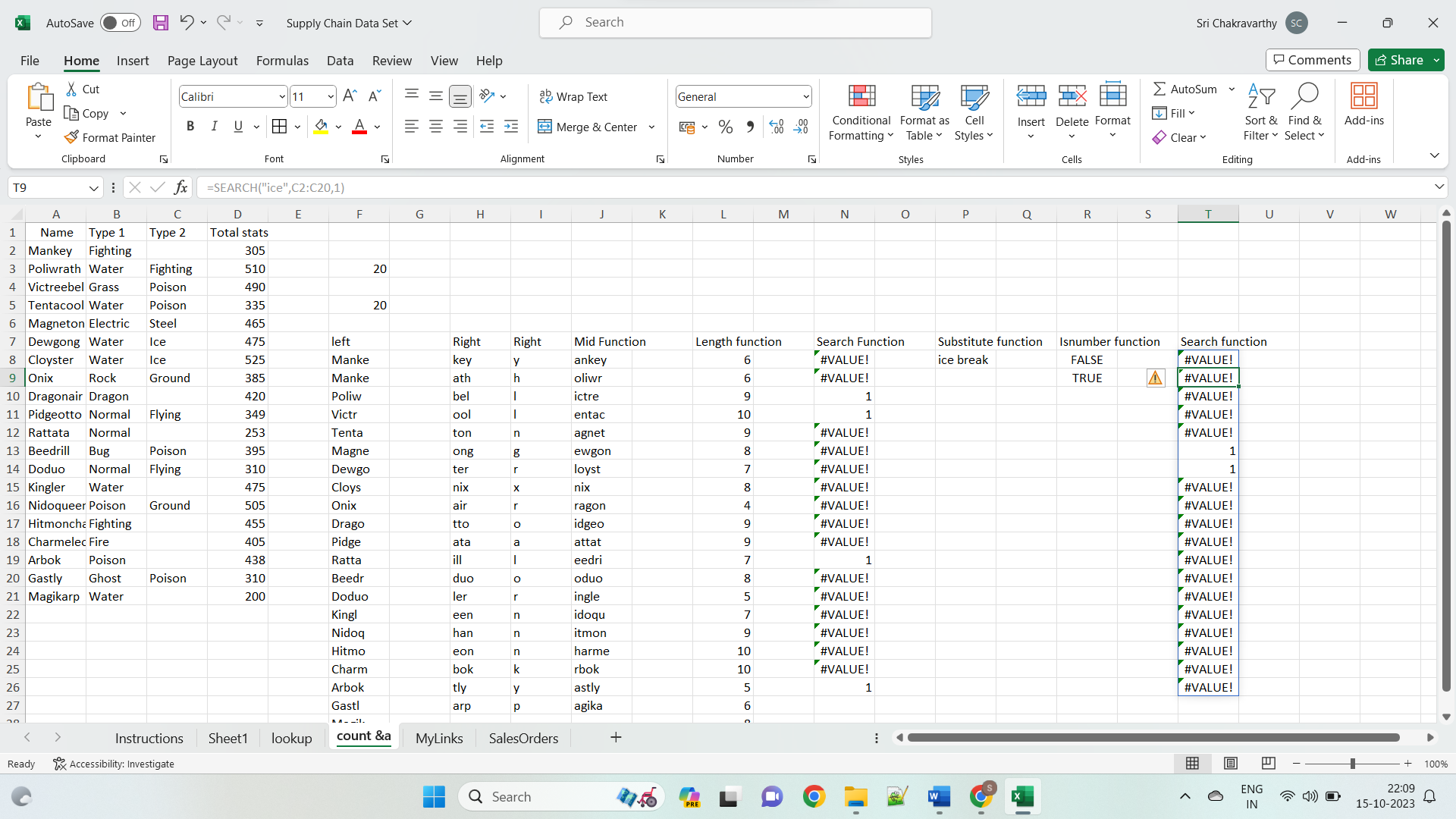
Excel SEARCH function has three-parameter two (find\_text, within\_text) are compulsory parameters and one (start\_num) is optional.

**Compulsory Parameter:**

* **find\_text:** find\_text refers to the substring/character you want to search within a string or the text you want to find out.
* **within\_text:**. Where your substring is located or where you perform the find\_text.

**Optional Parameter:**

* **[start\_num]::**from where you want to start the SEARCH within the text in excel. If omitted, SEARCH considers it as 1 and star search from the first character.



**Date Function:**

**DATE(year, month, day)** returns a serial number of a date based on the year, month and day values that you specify.

When it comes to working with dates in Excel, DATE is the most essential function to understand. The point is that other Excel date functions not always can recognize dates entered in the text format. So, when performing date calculations in Excel, you'd better supply dates using the DATE function to ensure the correct results.

Here are a few Excel DATE formula examples:

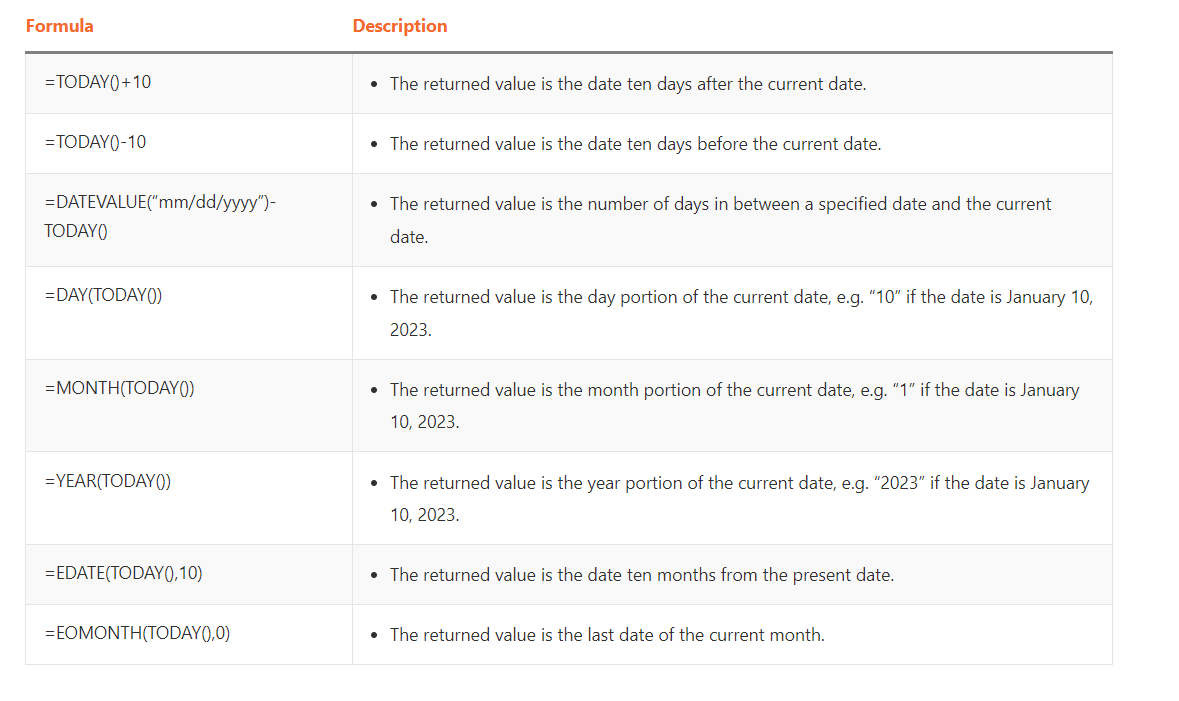
**=DATE(2023, 10, 29)** - returns a serial number corresponding to 29-OCT-2023.

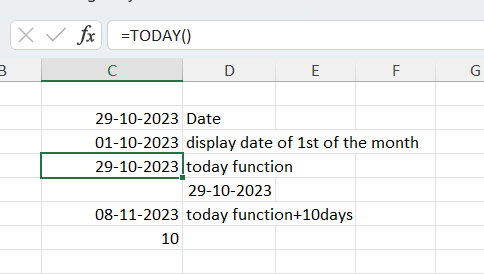
**=DATE(YEAR(TODAY()), MONTH(TODAY()), 1)** - returns the first day of the current year and month.

**=DATE(2015, 5, 20)-5** - subtracts 5 days from May 20, 2015.

Today function:

The Excel TODAY() function can be entered into a cell to return the current date. The output of the TODAY function date—the present date—is continuously updated to the date on which the file was most recently opened.

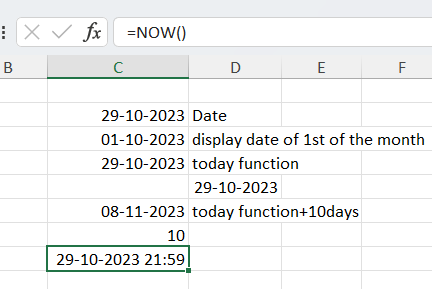




NOW function:

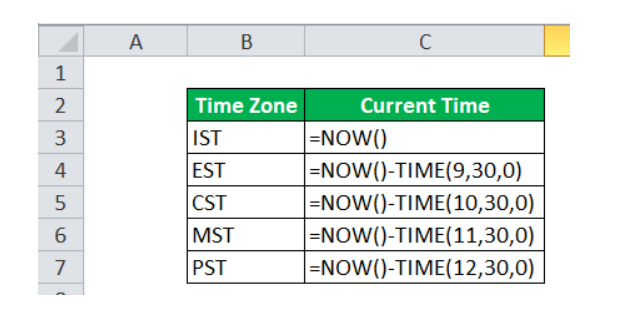
The NOW function in Excel is very useful when dealing with dates and times. The NOW Excel function is very simple and easy to use. Let us understand the working of =NOW()

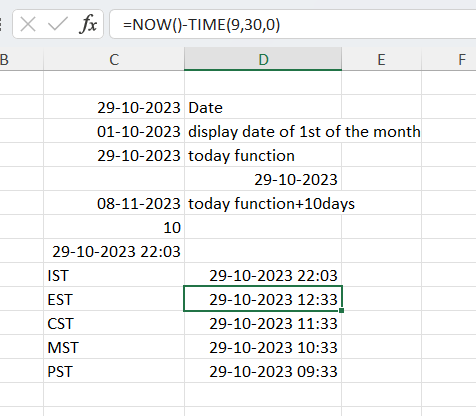
The NOW formula in Excel displays the current time as a time serial number (or a serial number without an associated date):



The main time zones are:

* Eastern Standard Time Zone (EST)
* Central Standard Time Zone (CST)
* Mountain Standard Time Zone (MST)
* Pacific Standard Time Zone (PST)





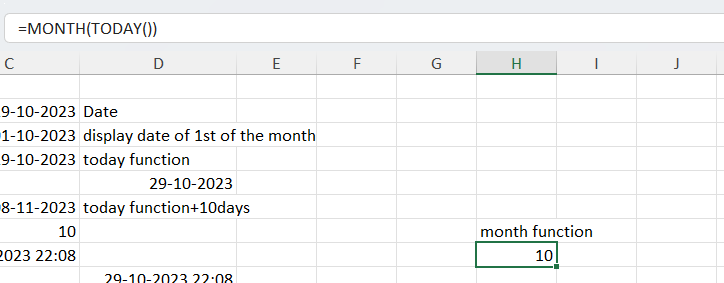
Month function:

**MONTH(serial\_number)** function in Excel returns the month of a specified date as an integer ranging from 1 (January) to 12 (December).

For example:

**=MONTH(A2)** - returns the month of a date in cell A2.

**=MONTH(TODAY())** - returns the current month.



Year function:

**YEAR(serial\_number)** returns a year corresponding to a given date, as a number from 1900 to 9999.

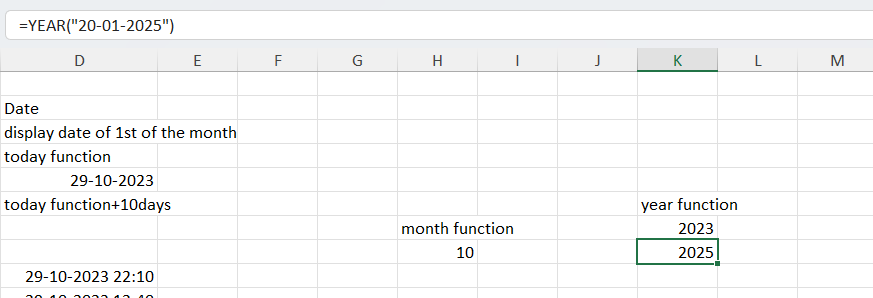
The Excel YEAR function is very straightforward and you will hardly run into any difficulties when using it in your date calculations:

**=YEAR(A2)** - returns the year of a date in cell A2.

**=YEAR("20-May-2023")** - returns the year of the specified date.

**=YEAR(DATE(2023,5,20))** - a more reliable method to get the year of a given date.

**=YEAR(TODAY())** - returns the current year.



EOmonth Function:

**EOMONTH(start\_date, months)** function returns the last day of the month a given number of months from the start date.

Like most of Excel date functions, EOMONTH can operate on dates input as cell references, entered by using the DATE function, or results of other formulas.

A **positive value** in the **months** argument adds the corresponding number of months to the start date, for example:

**=EOMONTH(A2, 3)** - returns the last day of the month, 3 months **after** the date in cell A2.

A **negative value** in the *months* argument subtracts the corresponding number of months from the start date:

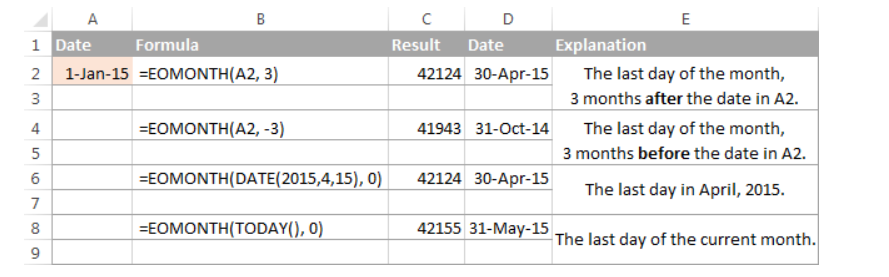
**=EOMONTH(A2, -3)** - returns the last day of the month, 3 months **before** the date in cell A2.

A **zero** in the *months* argument forces the EOMONTH function to return the last day of the start date's month:

**=EOMONTH(DATE(2015,4,15), 0)** - returns the last day in April, 2015.

To get the **last day of the current month**, enter the [TODAY function](https://www.ablebits.com/office-addins-blog/excel-date-functions/#TODAY) in the *start\_date* argument and 0 in *months*:

**=EOMONTH(TODAY(), 0)**



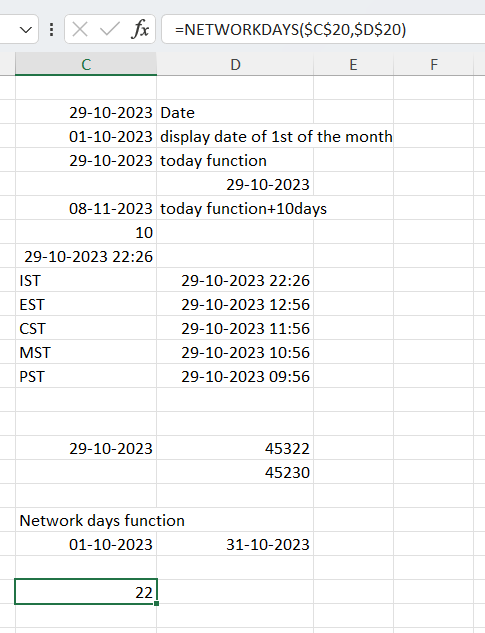
Networkdays function:

The NETWORKDAYS Function[[1]](https://corporatefinanceinstitute.com/resources/excel/calculate-working-days-excel-between-two-dates/#citation-networkdays-function-48e717bf-a7a3-495f-969e-5005e3eb18e7) calculates the number of workdays between two dates in Excel. When using the function, the number of weekends are automatically excluded. It also allows you to skip specified holidays and only count business days. It is categorized in Excel as a [Date/Time Function](https://corporatefinanceinstitute.com/topic/excel/).

**=NETWORKDAYS(start\_date, end\_date, [holidays])**

The function uses the following arguments:

1. **Start\_date** (required argument) – Start\_date can either be earlier than end\_date, later than end\_date, or same as end\_date.
2. **End\_date**(required argument) – The end date.
3. **Holidays**(optional argument) – This specifies the list of holidays that should be excluded from the work days calculation. We can enter it as a range of cells that contain the holiday dates (that is F2:F4) or as a list of serial numbers that represent the holiday dates.



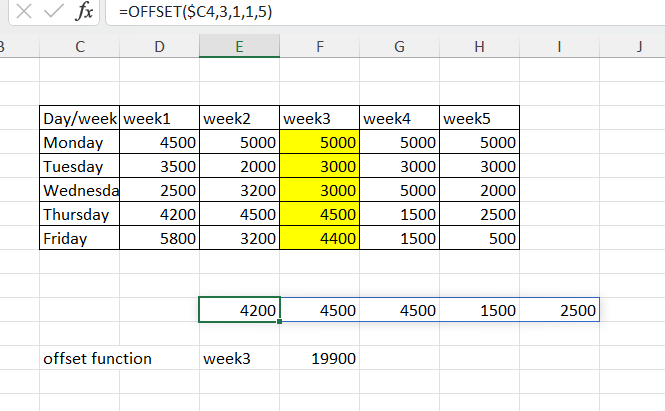
Offset:

The OFFSET Function is categorized under Excel Lookup and Reference functions. OFFSET will return a range of cells. That is, it will return a specified number of rows and columns from an initial range that was specified.

**= OFFSET(reference, rows, cols, [height], [width])**

The OFFSET function uses the following arguments:

1. **Reference** (required argument) – This is the cell range that is to be offset. It can be either single cell or multiple cells
2. **Rows** (required argument) – This is the number of rows from the start (upper left) of the supplied reference, to the start of the returned range.
3. **Cols**(required argument) – The number of columns from the start (upper left) of the supplied reference, to the start of the returned range.
4. **Height** (optional argument) – Specifies the height of the returned range. If omitted, the returned range is the same height as the supplied reference argument.
5. **Width** (optional argument) – This specifies the width of the returned range. If omitted, the returned range is the same width as the supplied reference.



Choose function:

The CHOOSE function in Excel is designed to return a value from the list based on a specified position.

The syntax of the CHOOSE function is as follows:

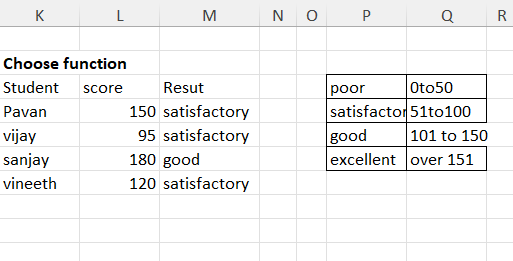
**CHOOSE(index\_num, value1, [value2], …)**

**ndex\_num** (required) - the position of the value to return. It can be any number between 1 and 254, a cell reference, or another formula.

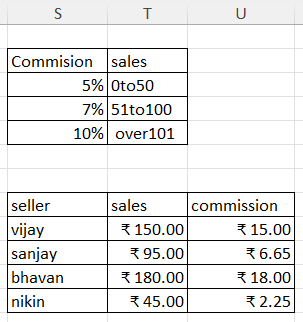
**Value1, value2, …** - a list of up to 254 values from which to choose. Value1 is required, other values are optional. These can be numbers, text values, cell references, formulas, or defined names.

Ex:

**=CHOOSE((L4>0)+(L4>=51)+(L4>=101)+(L4>=151),"poor","satisfactory","good","Excellent")**



EX: =CHOOSE((T9>=0)+(T9>=51)+(T9>=101),T9\*5%,T9\*7%,T9\*10%)



Let function:

The Excel LET function allows you to assign names to calculation results and define variables inside a formula, so that the formula looks clearer and works faster.

Essentially, the concept is the same as naming cells, ranges and formulas in the Name Manager. What makes the LET function different is that the declared names only exist in the scope of a given formula and nowhere else.

### **Syntax**

LET has the following syntax and arguments:

**LET (name1, name\_value1, [name2], [name\_value2], …, calculation)**

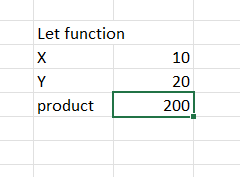
Where:

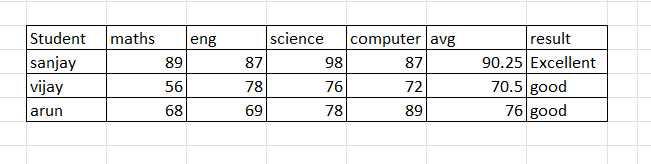
* **Name1** (required) - the first name to assign. It must begin with a letter.
* **Name\_value1** (required) - the value or calculation assigned to name1.
* **Name2 / name\_value2** (optional) - the second name and value.
* **Calculation** (required) - a calculation that uses assigned names and values.

**=LET(x, 2, y, 5, x\*y)**

**=LET(X,C16,Y,C17,X\*Y)**

**=LET(X,C16,Y,C17,(X+2\*X-Y))**

****

****

**=LET(avg,AVERAGE(I16:L16),IF(avg>90,"Excellent",IF(avg>50,"good",IF(avg>50,"satisfactory","poor"))))**

**Max function:**

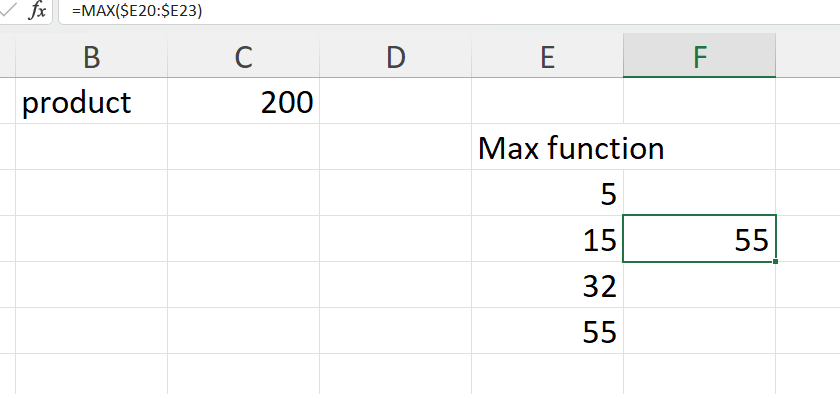
The MAX function in Excel returns the highest value in a set of data that you specify.

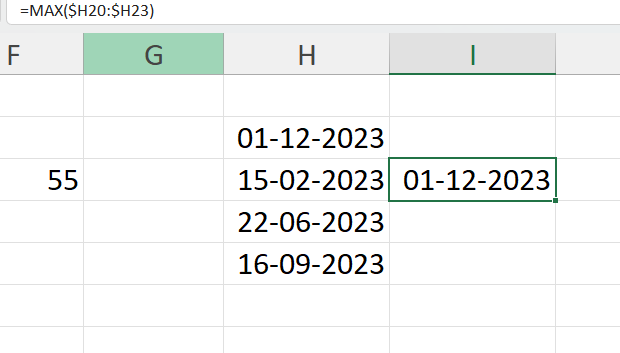
The syntax is as follows:

**MAX(number1, [number2], …)**

Where **number** can be represented by a numeric value, array, named range, a reference to a cell or range containing numbers.

*Number1* is required, *number2* and subsequent arguments are optional.

****

****

**Sort Function:**

The SORT function in Excel sorts the contents of an array or range by columns or rows, in ascending or descending order.

SORT belongs to the group of Dynamic array functions. The result is a dynamic array that automatically spills to neighbouring cells vertically or horizontally, depending on the shape of the source array.

The syntax of the SORT function is as follows:

**SORT(array, [sort\_index], [sort\_order], [by\_col])**

Where:

Array (required) - is an array of values or a range of cells to sort. These can be any values including text, numbers, dates, times, etc.

Sort\_index (optional) - an integer that indicates which column or row to sort by. If omitted, the default index 1 is used.

Sort\_order (optional) - defines the sort order:

* 1 or omitted (default) - ascending order, i.e. from smallest to largest
* -1 - descending order, i.e. from largest to smallest

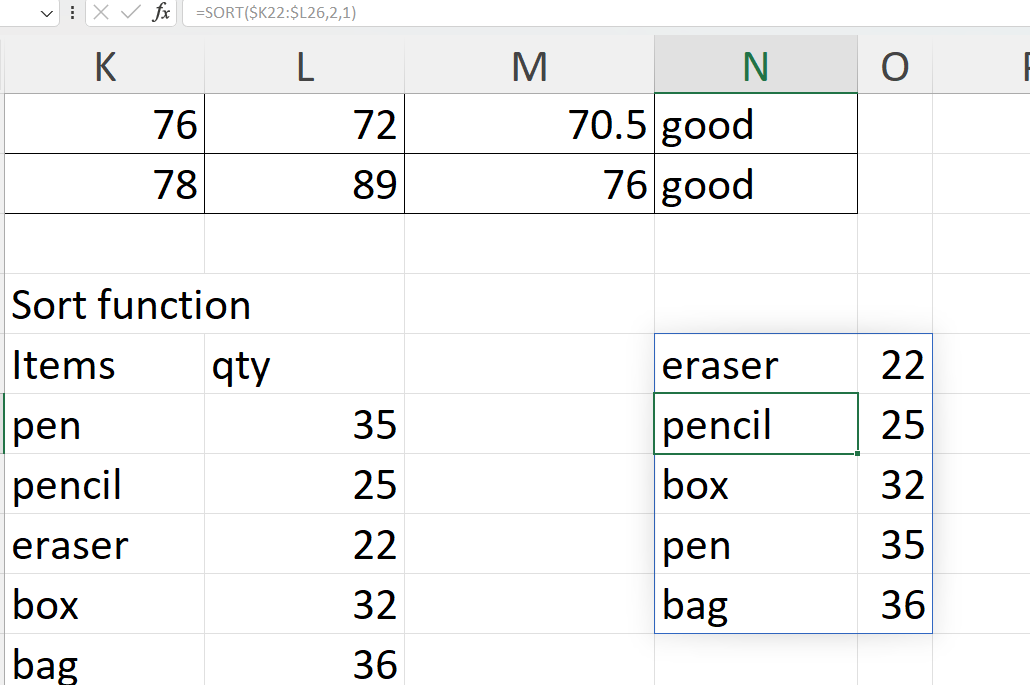
By\_col (optional) - a logical value that indicates the direction of sorting:

* FALSE or omitted (default) - sort by row. You'll use this option most of the time.
* TRUE - sort by column. Use this option if your data is organized horizontally in columns

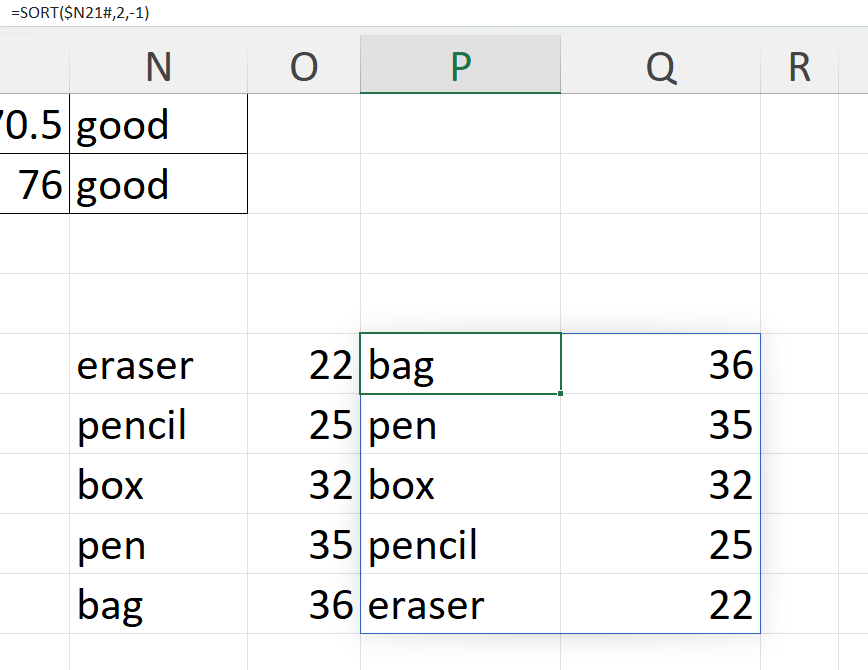
**=SORT(A2:B8, 2, 1)**

Where:

* A2:B8 is the source array
* 2 is the column number to sort by
* 1 is the ascending sort order



In descending:



Sortby function:

The SORTBY function in Excel is designed to sort one range or array based on the values in another range or array. Sorting can be done by one or multiple columns.

The SORTBY function has a variable number of arguments - the first two are required and the other are optional:

**SORTBY(array, by\_array1, [sort\_order1], [by\_array2, sort\_order2],…)**

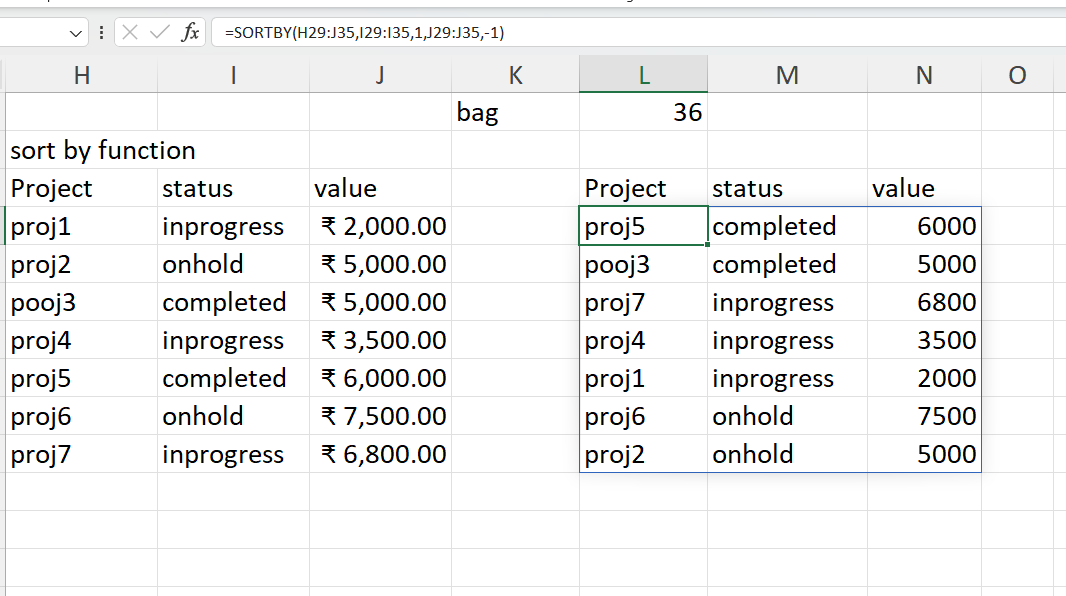
**Array** (required) - the range of cells or array of values to be sorted.

**By\_array1** (required) - the range or array to sort by.

**Sort\_order1** (optional) - the sorting order:

* 1 or omitted (default) - ascending
* -1 - descending

**By\_array2**/ **Sort\_order2**, … (optional) - additional array / order pairs to use for sorting.



Rank function:

The RANK function in Excel returns the order (or rank) of a numeric value compared to other values in the same list. In other words, it tells you which value is the highest, the second highest, etc.

The syntax of the Excel RANK function is as follows:

**RANK(number,ref,[order])**

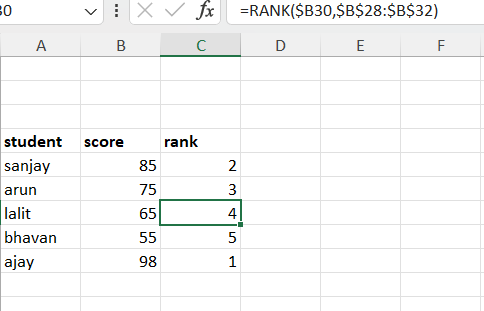
Where:

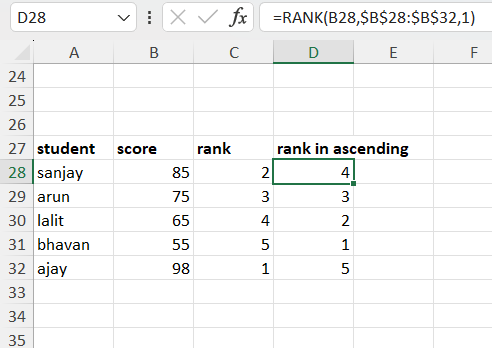
**Number** (required) - the value whose rank you'd like to find.

**Ref** (required) - a list of numeric values to rank against. It can be supplied as an array of numbers or a reference to the list of numbers.

**Order**(optional) - a number that specifies how to rank values:

* If 0 or omitted, the values are ranked in descending order, i.e. from largest to smallest.
* If 1 or any other non-zero value, the values are ranked in ascending order, i.e. from smallest to largest.





Filter function:

The FILTER function in Excel is used to filter a range of data based on the criteria that you specify.

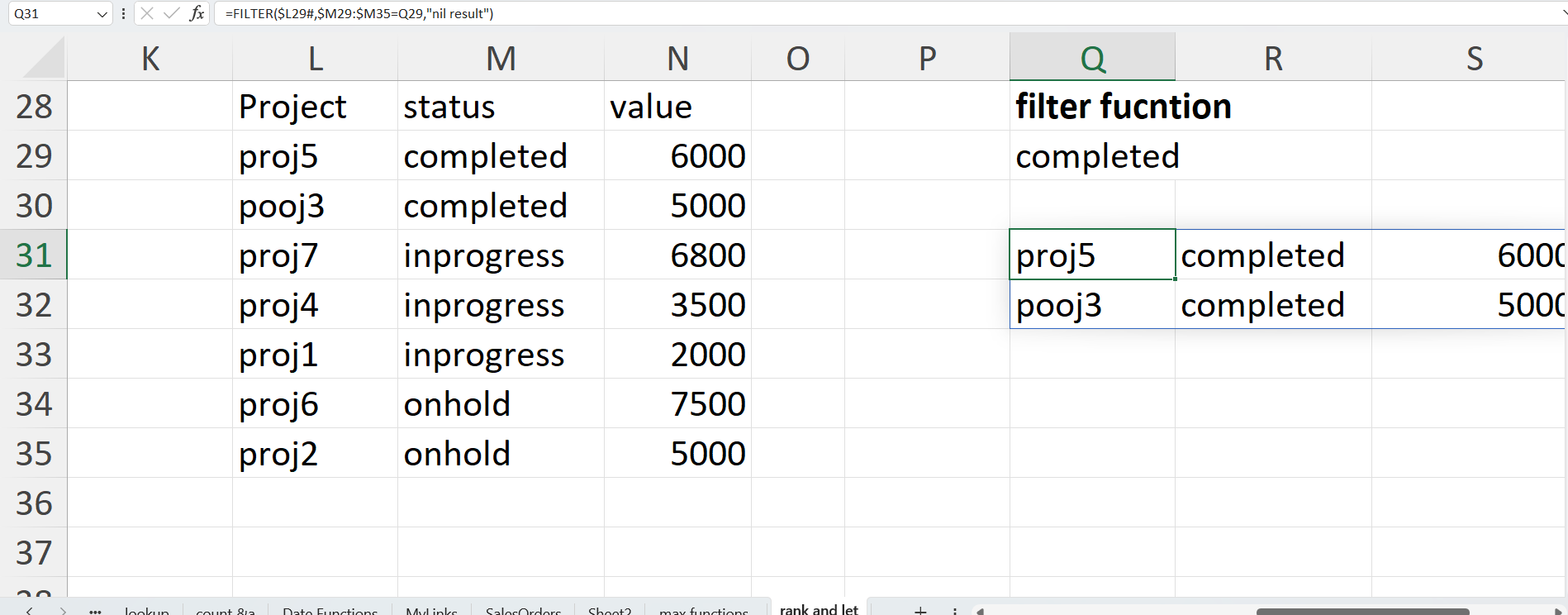
The function belongs to the category of Dynamic Arrays functions. The result is an array of values that automatically spills into a range of cells, starting from the cell where you enter a formula.

The syntax of the FILTER function is as follows:

**FILTER(array, include, [if\_empty])**

Where:

* **Array**(required) - the range or array of values that you want to filter.
* **Include** (required) - the criteria supplied as a Boolean array (TRUE and FALSE values).  
  Its height (when data is in columns) or width (when data is in rows) must be equal to the that of the array argument.
* **If\_empty** (optional) - the value to return when no entries meet the criteria.
* **Ex:** **=FILTER($L29#,$M29:$M35=Q29,"nil result")**



To extract cells that contain certain text, you can use the FILTER function together with the classic If cell contains formula:

**FILTER(array, ISNUMBER(SEARCH("text", range)), "No results")**

**Ex: =FILTER(L29#,ISNUMBER(SEARCH(S29,M29:M35)),"no result")**

****

**Frequency function:**

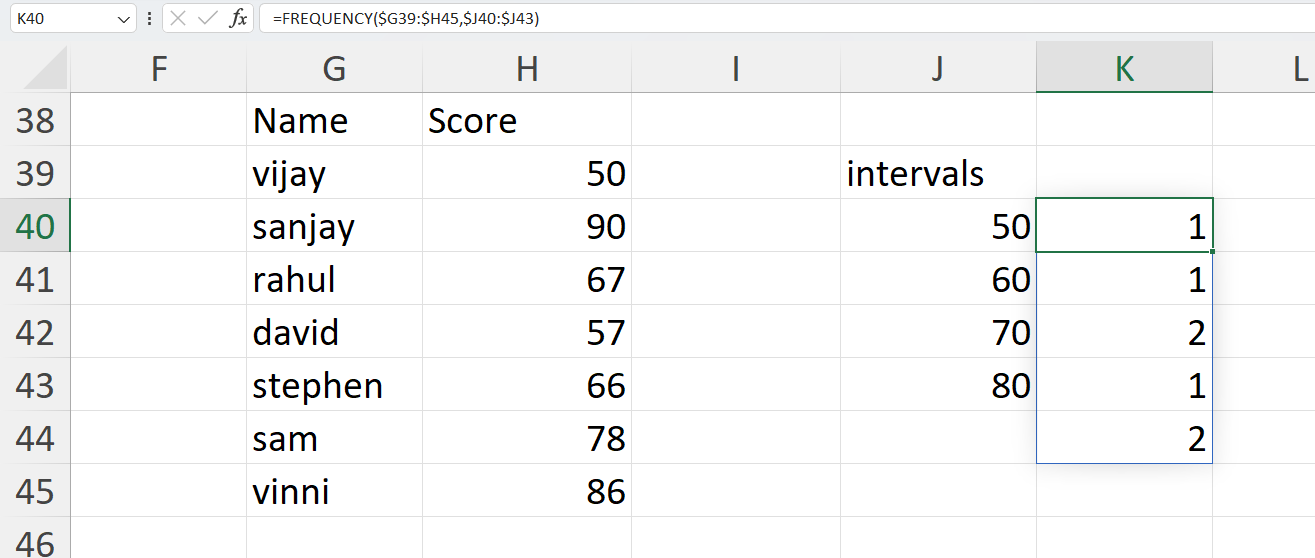
The FREQUENCY function returns how often values occur within a set of data. It returns a vertical array of numbers.

**Syntax: FREQUENCY( data\_array, bins\_array )**

where the function arguments are as follows:

|  |  |  |
| --- | --- | --- |
| data\_array | - | The original array of values that the frequency is to be calculated for. |
| bins\_array | - | An array of values, that specify the limits of the ranges that the data\_array is to be split into. |

**Ex: =FREQUENCY($G39:$H45,$J40:$J43)**

****

**Sequence function:**

The SEQUENCE function in Excel is used to generate an array of sequential numbers such as 1, 2, 3, etc.

It is a new dynamic array function introduced in Microsoft Excel 365. The result is a dynamic array that spills into the specified number of rows and columns automatically.

The function has the following syntax:

**SEQUENCE(rows, [columns], [start], [step])**

Where:

**Rows** (optional) - the number of rows to fill.

**Columns** (optional) - the number of columns to fill. If omitted, defaults to 1 column.

**Start** (optional) - the starting number in the sequence. If omitted, defaults to 1.

**Step** (optional) - the increment for each subsequent value in the sequence. It can be positive or negative.

* If positive, subsequent values increase, creating an ascending sequence.
* If negative, subsequent values decrease, producing a descending sequence.
* If omitted, the step defaults to 1.

To put numbers in a **column**:

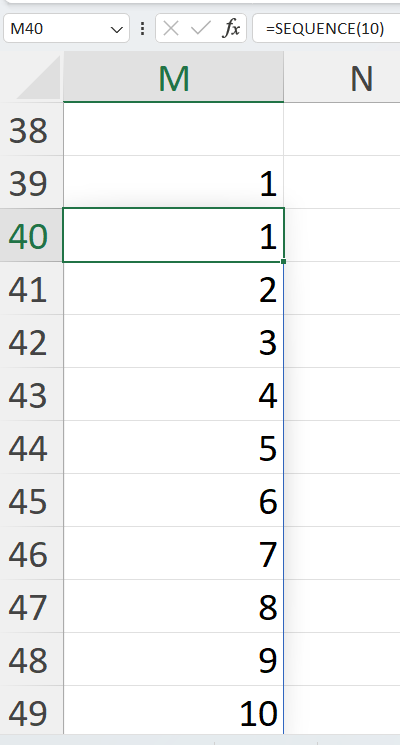
**SEQUENCE(*n*)**

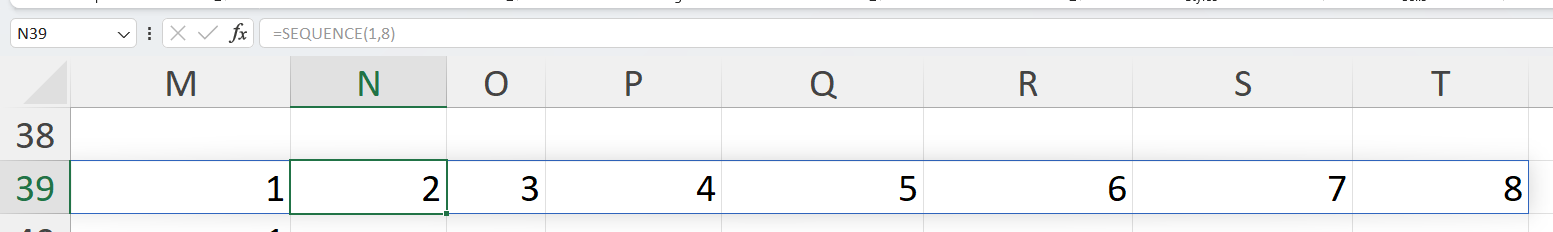
**Ex: sequence(10)**

To place numbers in a **row**:

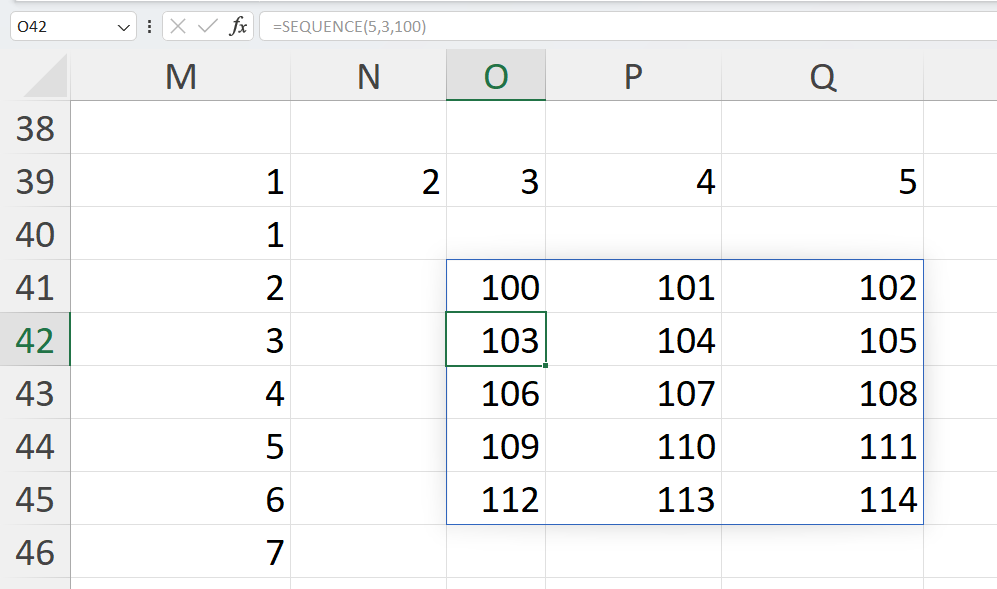
**SEQUENCE(1, *n*)**

**Ex: sequence(1,8)**





Ex: sequence(5,3,100)



**RANDARRAY function:**

The RANDARRAY function in Excel returns an array of random numbers between any two numbers that you specify.

The function has the following syntax. Please notice that all the arguments are optional:

**RANDARRAY([rows], [columns], [min], [max], [whole\_number])**

Where:

**Rows** (optional) - defines how many rows to fill. If omitted, defaults to 1 row.

**Columns** (optional) - defines how many columns to fill. If omitted, defaults to 1 column.

**Min** (optional) - the smallest random number to produce. If not specified, the default 0 value is used.

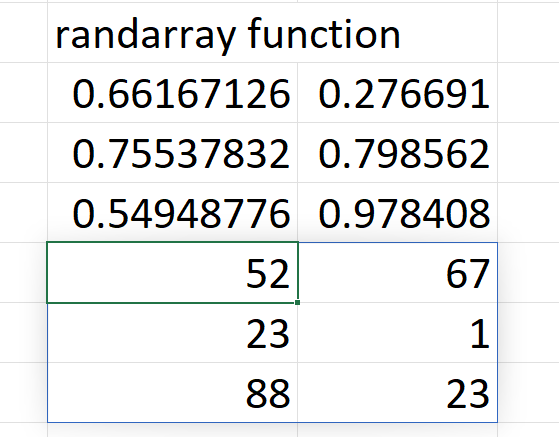
**Max** (optional) - the largest random number to create. If not specified, the default 1 value is used.

**Whole\_number** (optional) - determines what kind of values to return:

* TRUE - whole numbers
* FALSE or omitted (default) - decimal numbers

Ex: =RANDARRAY(3,2,1,100,TRUE)

=RANDARRAY(3,2)



**IFERROR function:**

The IFERROR function in Excel is designed to trap and manage errors in formulas and calculations. More specifically, IFERROR checks a formula, and if it evaluates to an error, returns another value you specify; otherwise, returns the result of the formula.

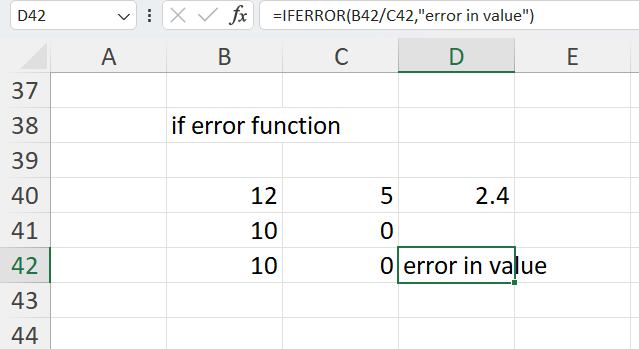
The syntax of the Excel IFERROR function is as follows:

**IFERROR(value, value\_if\_error)**

Where:

* **Value** (required) - what to check for errors. It can be a formula, expression, value, or cell reference.
* **Value\_if\_error** (required) - what to return if an error is found. It can be an empty string (blank cell), text message, numeric value, another formula or calculation.

**Ex:** **=IFERROR(B42/C42,"error in value")**



Pivot tables :

What is a Pivot Table in Excel?

An Excel *Pivot Table* is a tool to explore and summarize large amounts of data, analyze related totals and present summary reports designed to:

* Present large amounts of data in a user-friendly way.
* Summarize data by categories and subcategories.
* Filter, group, sort and conditionally format different subsets of data so that you can focus on the most relevant information.
* Rotate rows to columns or columns to rows (which is called "pivoting") to view different summaries of the source data.
* Subtotal and aggregate numeric data in the spreadsheet.
* Expand or collapse the levels of data and drill down to see the details behind any total.
* Present concise and attractive online of your data or printed reports.

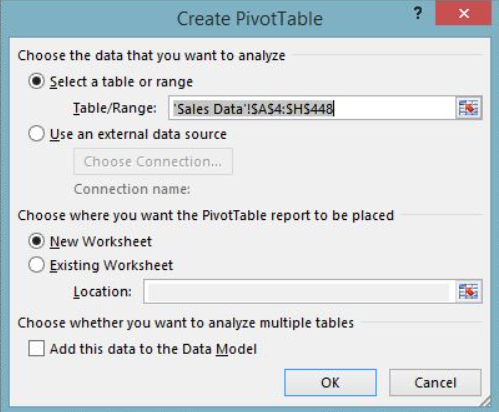
Insert Pivot Tables

To insert a pivot table in your sheet, follow these steps:

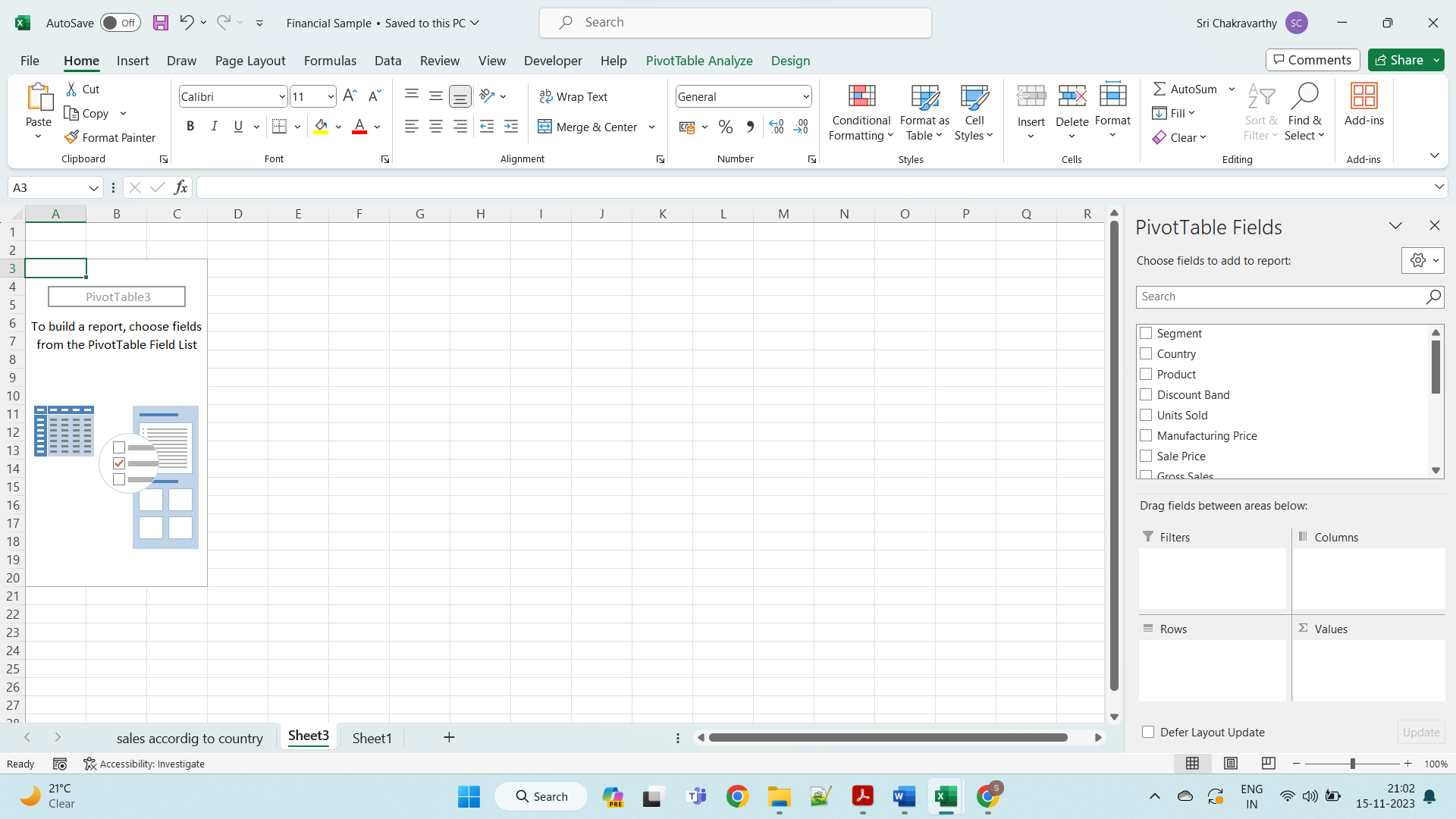
* Click on any cell in a data set.
* On the Insert tab, in the Tables group, click PivotTable.



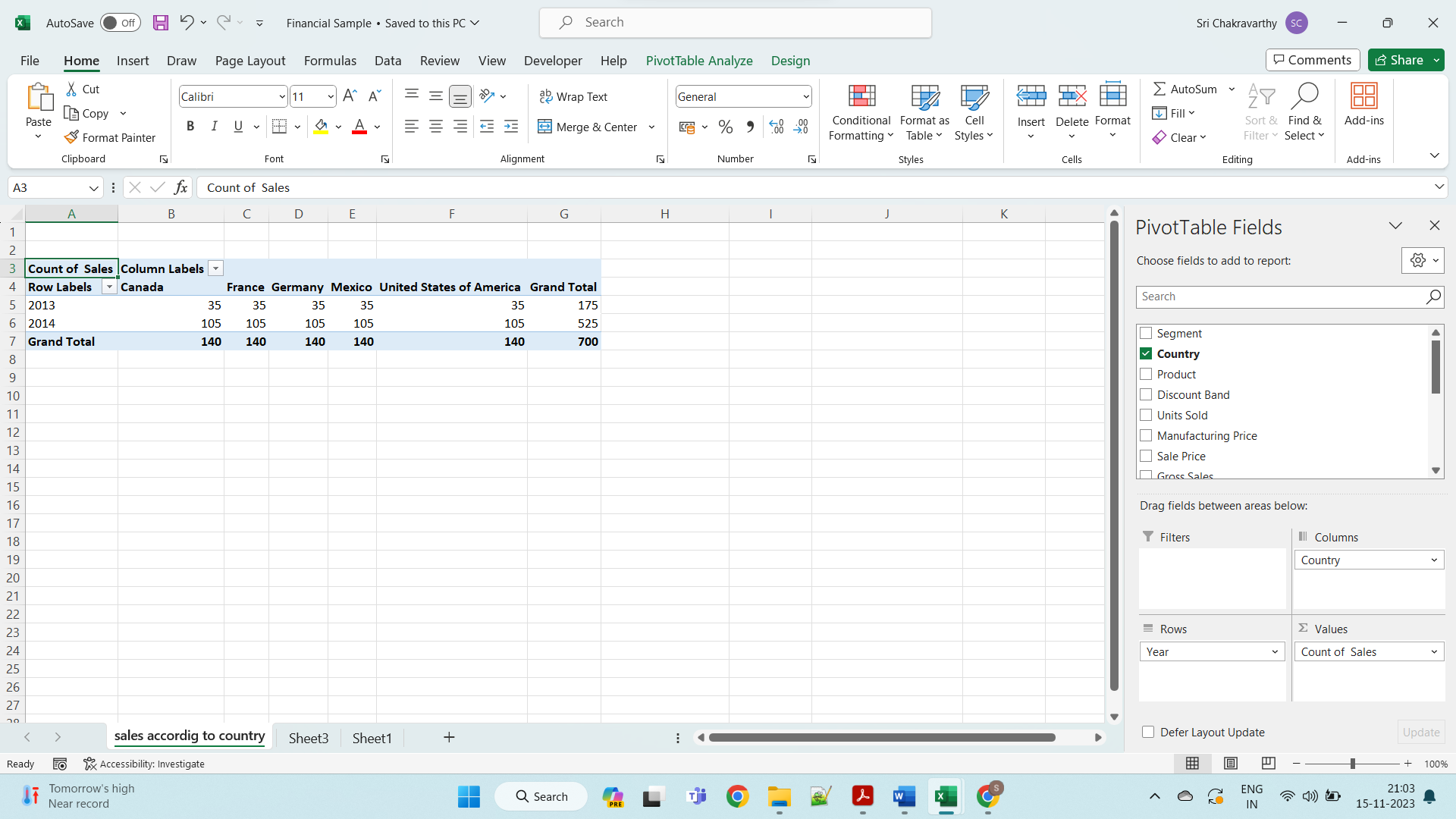
A dialog box will appear. Excel will auto-select your dataset. It will also create a new worksheet for your pivot table



* Click Ok. Then, it will create a pivot table worksheet.

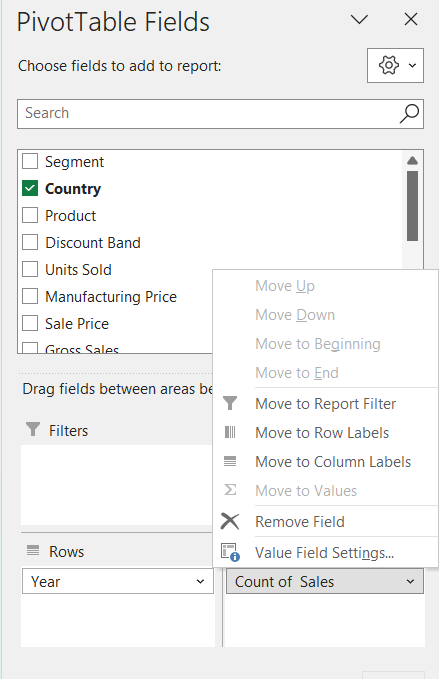


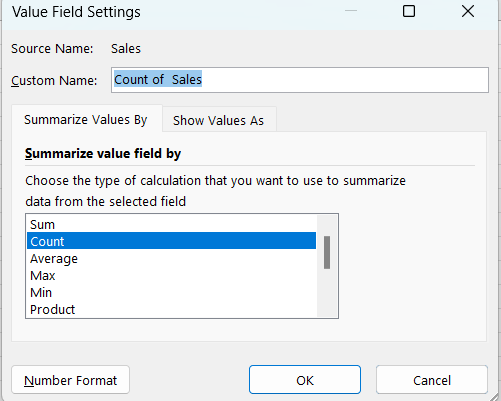
Drag the fields to the concern areas



## Value Field Settings

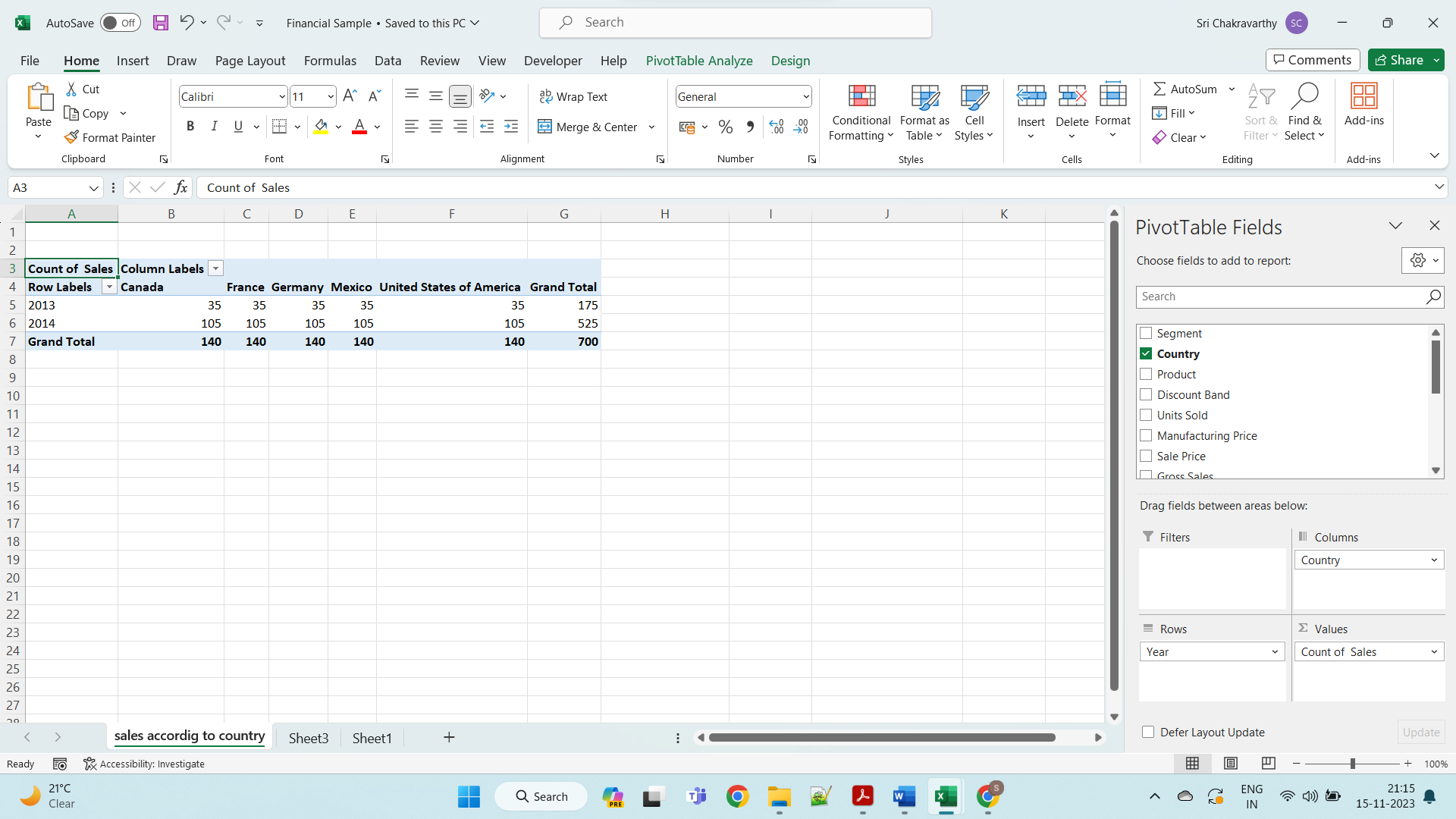
By default, Excel gives the summation of the values that are put into the Values section. You can change that from the Value Field Settings.





* Click OK.

## Two-Dimensional Pivot Table

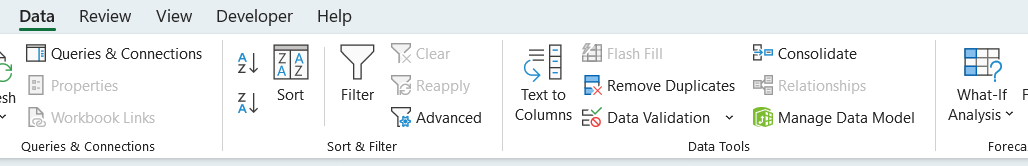
We can create a pivot table in various two-dimensional arrangements. Drag the following fields to the different areas

WHAT IF ANALYSIS :

In Excel, What-if analysis is a process of changing cells' values to see how those changes will affect the worksheet's outcome. You can use several different sets of values to explore all the different results in one or more formulas.

What-if Excel is used by almost every data analyst and especially middle to higher management professionals to make better, faster and more accurate decisions based on data. What-if analysis is useful in many situations, such as:

To enable the what-if analysis tool go to the **Data** menu tab and click on the What-If Analysis option under the Forecast section.



Now click on the **What-If Analysis**. [Excel](https://www.javatpoint.com/excel-tutorial) has the following What-if analysis tools that can be used based on the data analysis needs:

* Scenario Manager
* Goal Seek
* Data Tables

**1. Scenario Manager**

A scenario is a set of values that Excel saves and can substitute automatically in cells on a worksheet. Below are the following key features, such as:

* You can create and save different groups of values on a worksheet and then switch to any of these new scenarios to view different results.
* A scenario can have multiple variables, but it can accommodate only up to 32 values.
* You can also create a scenario summary report, which combines all the scenarios on one worksheet.

**Initial Values for Scenarios**

**Step 1:** Define the cells that contain the input values.

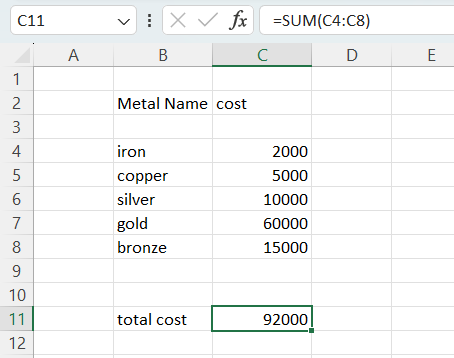
**Step 2:** Name the cells **Metals\_name** and **Cost**.

**Step 3:** Define the cells that contain the results.

**Step 4:** Name the result cell **Total\_cost**.

**Step 5:** place the formula in the result cell.

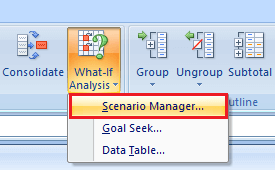
**Step 6:** Below is the created table.



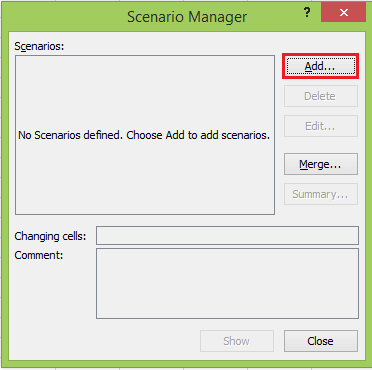
To create an analysis report with Scenario Manager, follow the following steps, such as:

**Step 1:** Click the ***Data*** tab.

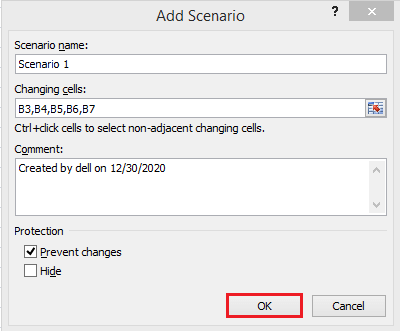
**Step2:** Go to the ***What-If Analysis*** button and click on the ***Scenario Manager*** from the dropdown list.



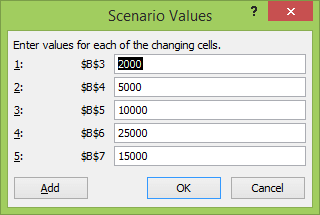
**Step 3:** Now a scenario manager dialog box appears, click on the ***Add*** button to create a scenario.



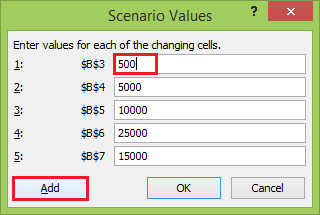
**Step 4:** Create the scenario, name the scenario, enter the value for each changing input cell for that scenario, and then click the ***Ok*** button.



**Step 5**: Now, B3, B4, B5, B6, and B7 appear in the cells box.



**Step 6:** Now, change the value of B3to 500 and click the ***Add*** button.

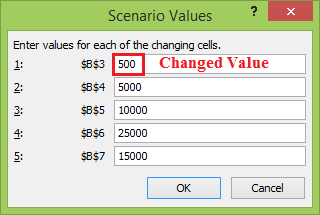


**Step 7:** After clicking on the Add button, the add scenario dialog box appears again.

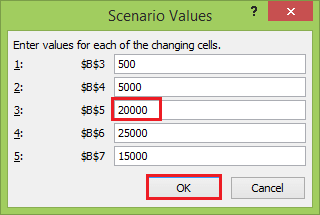
* In the scenario name box, create scenario 2.
* Select the prevent changes.
* And click on the ***Ok***



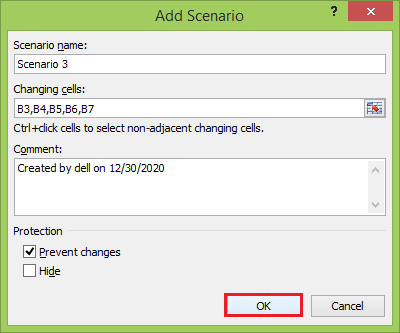
**Step 8:** Again appears scenario values box with the changed value of B3 cell.



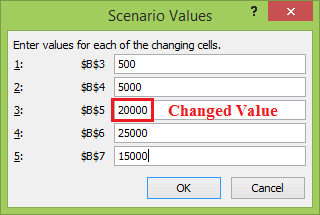
**Step 9:** Change the value of B5 to 20000 and click the ***Ok*** button.



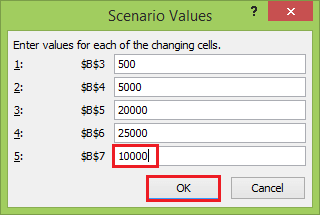
**Step 10:** Similarly, create Scenario 3 and click the ***Ok*** button.



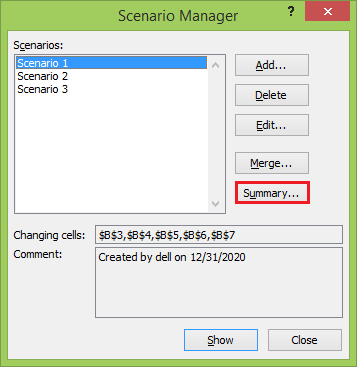
**Step 11:** Again, appears scenario values box with a changed value of the B5 cell.



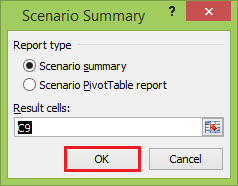
**Step 12:** Change the value of B7 to 10000 and click the ***Ok*** button.



The ***Scenario Manager*** Dialog box appears. In the box under Scenarios, You will find the names of all the scenarios that you have created.



**Step 13:** Now, click on the ***Summary*** button. The Scenario Summary dialog box appears.



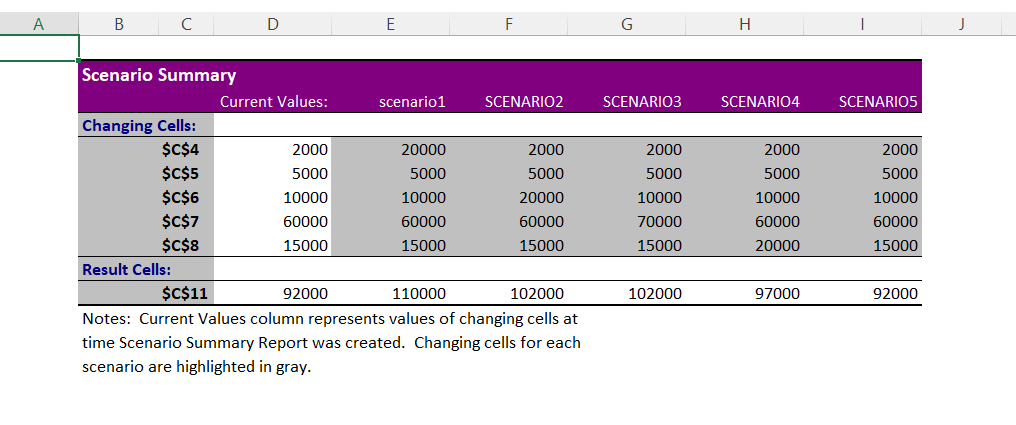
Excel provides two types of Scenario Summary reports:

1. Scenario summary.
2. Scenario PivotTable report.

**Step 14:** Select ***Scenario summary*** under Report type and click ***Ok***. Scenario Summary report appears in a new worksheet. You will get the following Scenario summary report.

ou can observe the following in the Scenario Summary report:

* **Changing Cells:** Enlists all the cells used as changing cells.
* **Result Cells:** Displays the result cell specified.
* **Current Values:** It is the first column and enlists the values of that scenario selected in the Scenario Manager Dialog box before creating the summary report.



Goal seek :

Goal Seek is a What-If Analysis tool that helps you to find the input value that results in a target value that you want. **Goal Seek** requires a formula that uses the input value to give the result in the target value. Then, by varying the formula's input value, Goal Seek tries to solve the input value.

**Step 1:** On the **Data** tab, go **What-If Analysis** and click on the **Goal Seek** option.



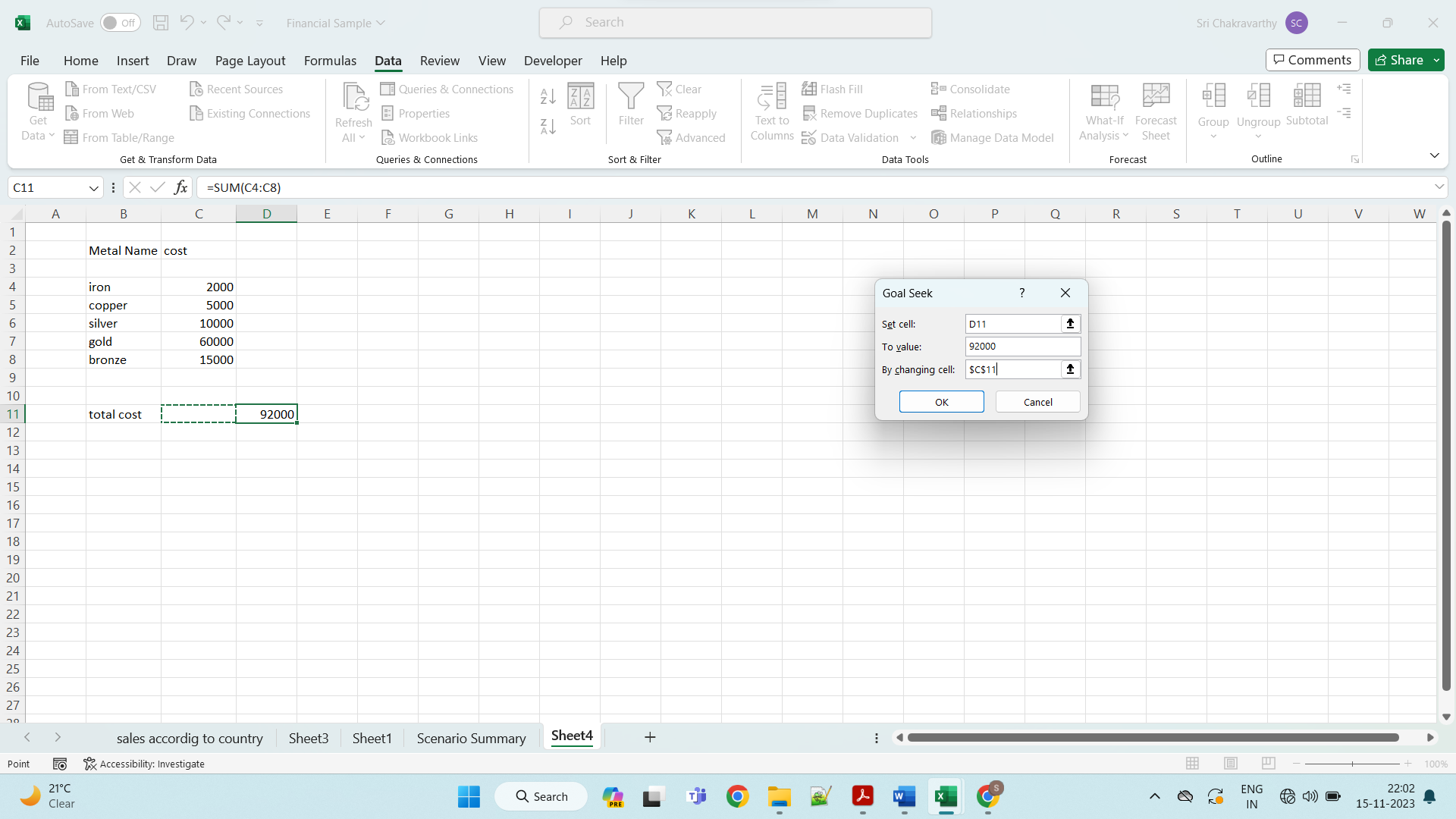
**Step 2:** The ***Goal Seek*** dialog box appears.

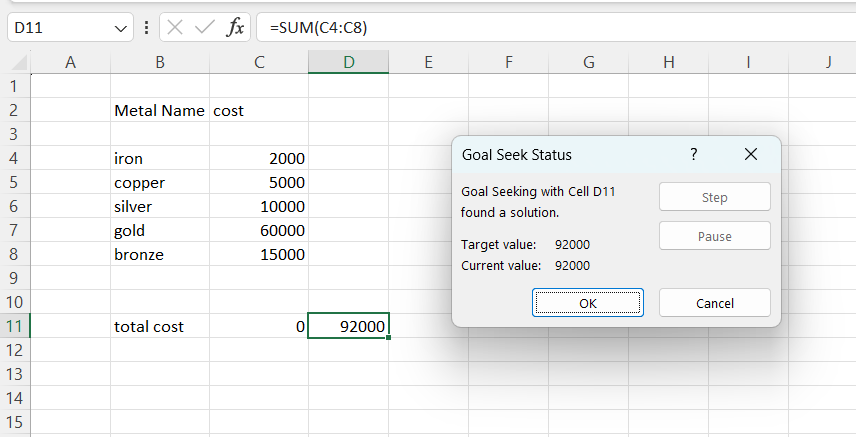
**Step 3:** Type D11 in the ***Set cell*** box. This box is the reference for the cell that contains the formula that you want to resolve.

**Step 4:** Type 92000 in the ***To value*** box. Here, you get the formula result.

**Step 5:** Type c11 in the ***By changing cell*** box. This box has the reference of the cell that contains the value you want to adjust.

**Step 6:** This cell that the formula must reference goal Seek changes in the cell that you specified in the Set cell box. Click ***Ok***.





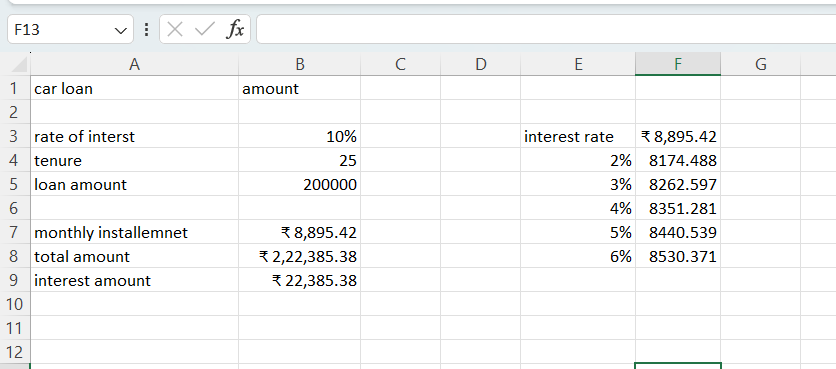
### **What-If Analysis Data Tables**

With a Data Table in Excel, you can easily vary one or two inputs and perform a What-if analysis. A Data Table is a range of cells where you can change values in some of the cells and answer different answers to a problem. There are two types of Data Tables, such as:

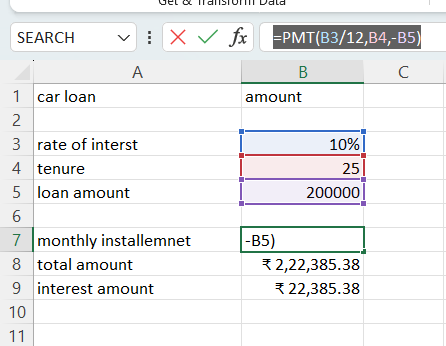
1. One-variable data tables
2. Two-variable data tables

If you have more than two variables in your analysis problem, you need to use the Excel Scenario Manager Tool.

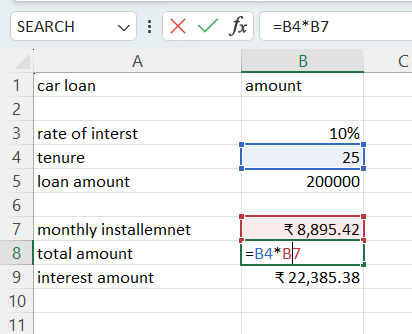
**One-variable Data Tables**



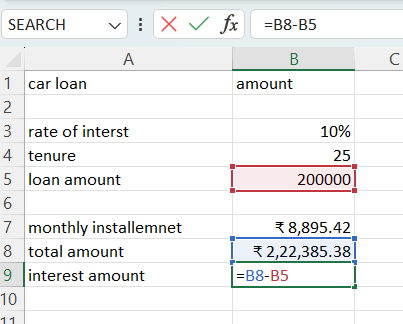
To calculate the emi us ethe formula as



Total amount:

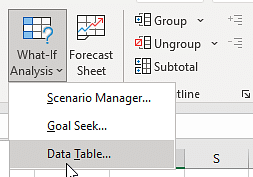


Interest amount:

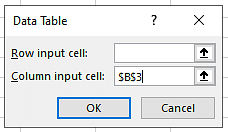


Now, create a new column with all the rates of interest you wish to look for. Followed by that, write the cell address of the EMI in your table in the next column. In this case, it is B7. The table should appear something like below.

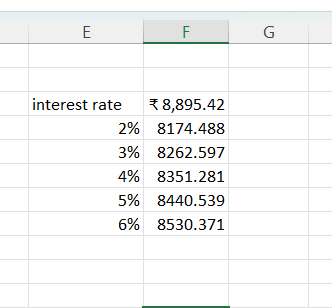
Now, select the entire table, click on the Excel What-if Analysis, and select the data table option.



A new window will pop up on the screen. In the dialog box, provide the rate of the interest cell address. In this case, the cell address is B3. The image will appear as follows.



Now, select OK, and the final data will appear as shown below.



Data Validation:

**Excel Data Validation** is a feature that restricts (validates) user input to a worksheet. Technically, you create a validation rule that controls what kind of data can be entered into a certain cell.

Here are just a few examples of what Excel's data validation can do:

* Allow only **numeric** or **text** values in a cell.
* Allow only numbers within a specified **range**.
* Allow data entries of a specific **length**.
* Restrict dates and times outside a given **range**.
* Restrict entries to a selection from a **drop-down list**.
* Validate an entry based on **another cell**.
* Show an **input message** when the user selects a cell.
* Show a **warning message** when incorrect data has been entered.
* Find **incorrect entries** in validated cells.

## How to do data validation in Excel

To add data validation in Excel, perform the following steps.

#### **1. Open the Data Validation dialog box**

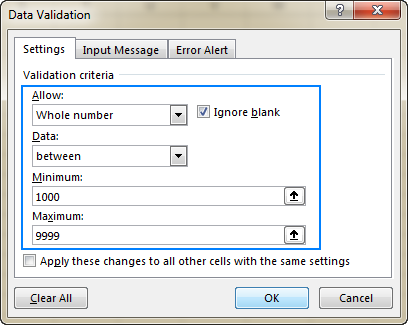
Select one or more cells to validate, go to the Data tab > Data Tools group, and click the **Data Validation** button.

You can also open the Data Validation dialog box by pressing Alt > D > L, with each key pressed separately.  


#### **2. Create an Excel validation rule**

On the **Settings** tab, define the validation criteria according to your needs. In the criteria, you can supply any of the following:

* Values - type numbers in the criteria boxes like shown in the screenshot below.
* Cell references - make a rule based on a value or formula in [another cell](https://www.ablebits.com/office-addins-blog/data-validation-excel/#validation-another-cell).
* Formulas - allow to express more complex conditions like in [this example](https://www.ablebits.com/office-addins-blog/data-validation-excel/#formula-based-validation-rules).

As an example, let's make a rule that restricts users to entering a whole number between 1000 and 9999:  


#### **3. Add an input message (optional)**

If you want to display a message that explains to the user what data is allowed in a given cell, open the Input Message tab and do the following:

* Make sure the **Show input message when cell is selected** box is checked.
* Enter the title and text of your message into the corresponding fields.
* Click OK to close the dialog window.



#### **4. Display an error alert (optional)**

In addition to the input message, you can show one of the following error alerts when invalid data is entered in a cell.

| **Alert type** | **Description** |
| --- | --- |
| **Stop** (default) | Stop alert The strictest alert type that prevents users from entering invalid data.  You click Retry to type a different value or Cancel to remove the entry. |
| **Warning** | Warning alert Warns users that the data is invalid, but does not prevent entering it.  You click Yes to input the invalid entry, No to edit it, or Cancel to remove the entry. |
| **Information** | Information alert The most permissive alert type that only informs users about an invalid data entry.  You click OK to enter the invalid value or Cancel to remove it from the cell. |

How to copy Excel data validation rule to other cells

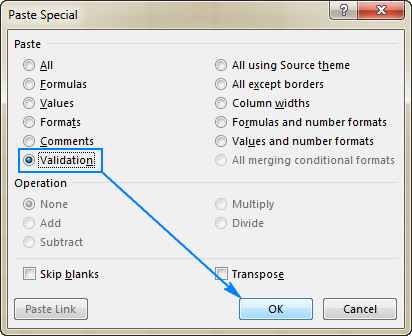
If you've configured data validation for one cell and wish to validate other cells with the same criteria, you don't have to re-create the rule from scratch.

To copy the validation rule in Excel, perform these 4 quick steps:

1. Select the cell to which the validation rule applies and press Ctrl + C to copy it.
2. Select other cells you want to validate. To select non-adjacent cells, press and hold the Ctrl key while selecting the cells.
3. Right-click the selection, click **Paste Special**, and then select the **Validation** option.

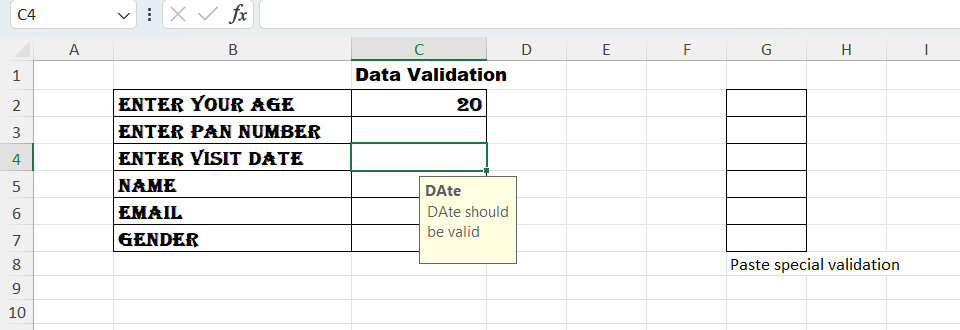
Alternatively, press the *Paste Special* > *Validation* shortcut: Ctrl + Alt + V, then N.

1. Click *OK*.



## How to find cells with data validation in Excel

To quickly locate all validated cells in the current worksheet, go to the Home tab > Editing group, and click Find & Select > **Data Validation**:  

**Subtotal with Ranges:**

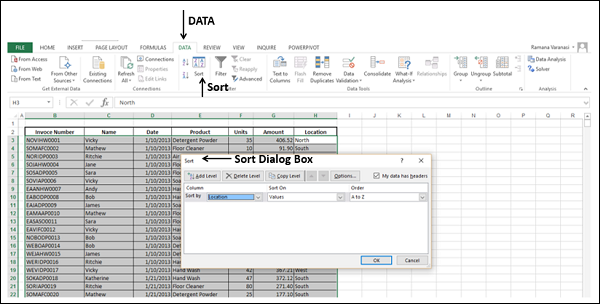
## Subtotals

You can obtain the sum of sales location wise using **Subtotal**.

First, sort the data location wise.

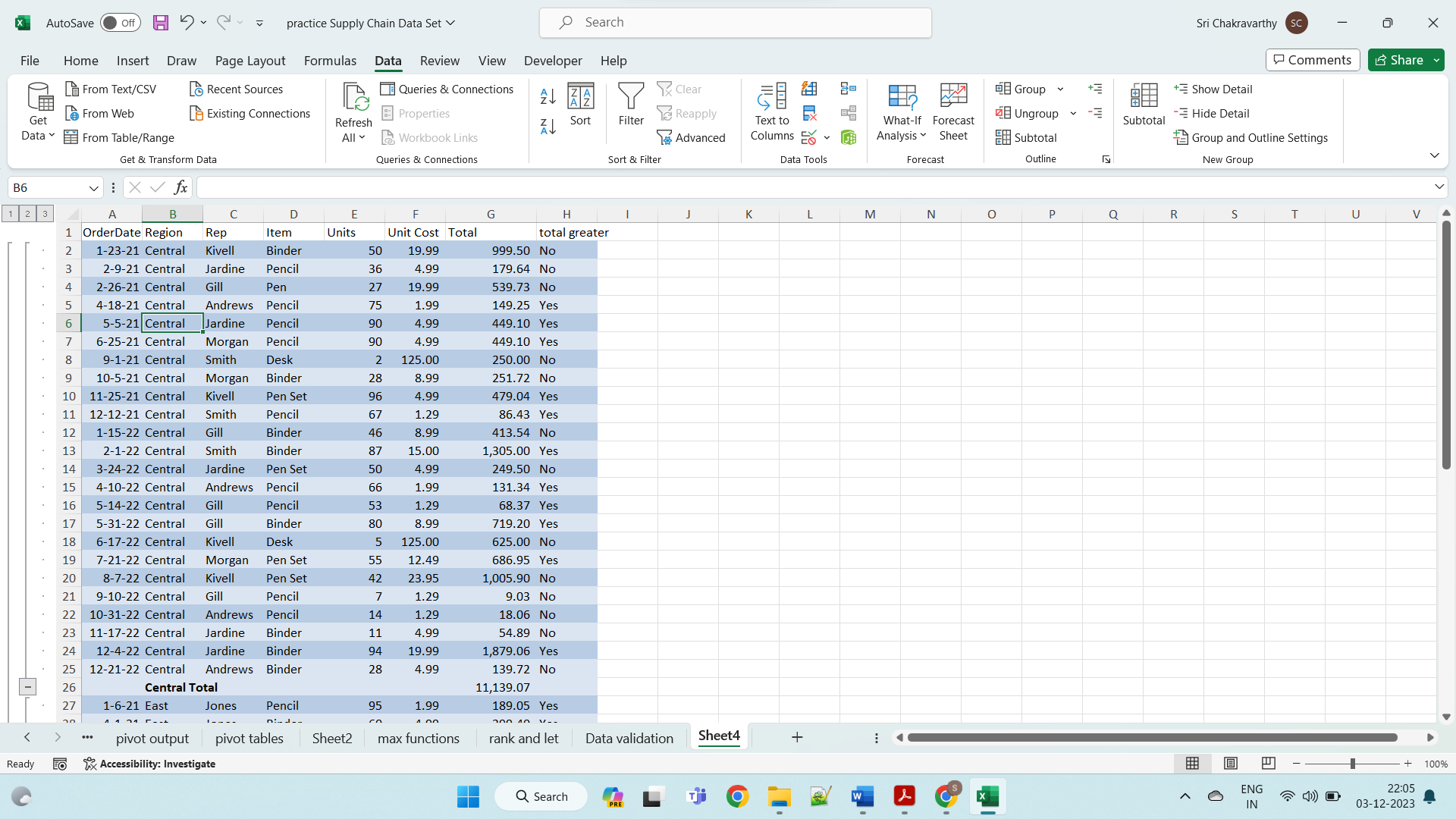
* Click anywhere on the data range.
* Click the **DATA** tab.
* Click **Sort**.

Data is selected. The **Sort** dialog box appears.



In the **Sort** dialog box,

* Select Location for **Sort by**
* Select Values for **Sort On**
* Select A to Z for **Order**



* Click anywhere on the Data Range.
* Click **DATA** tab.
* Click **Subtotal** in the **Outline** group. The data gets selected and the **Subtotal** dialog box appears.

