CA-C24L: Data Analytics Lab
Part A: Spreadsheet (Excel)

Data Set = Sample Super Store

1. Conditional Formatting, IF, COUNTIF, SUMIF, AVERAGE, CONCAT

a) Conditional Formatting

Explanation: Conditional formatting is used to change the appearance of cells in a range based on your specified **conditions**.
The conditions are rules based on specified numerical values or matching text.
The browser version of Excel provides a number of built-in conditions and appearances:

i) Conditional formatting on Sample Super store data set
Apply the Conditional formatting on profit column as follows:
  i) Select the entire profit column
  ii) Click on Conditional Formatting button from home tab

iii) Select Highlight Cell Rules option → Select Greater than option from it → Enter the value in the given text box for which profit you want to highlight the cells.

|	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	City	State	Postal Code	Region	Product ID	Category	Sub-Categor	Product Nan	Sales	Quantity	Discount	Profit		
2	Henderson	Kentucky	42420	South	FUR-BO-1000	Furniture	Bookcases	Bush Somers	261.96	2	0	41.9136	#NAME?	
3	Henderson	Kentucky	42420	South	FUR-CH-1000	Furniture	Chairs	Hon Deluxe F	731.94	3	0	219.582		
4	Los Angeles	California	90036	West	OFF-LA-1000	Office Suppli	Labels	Self-Adhesive	14.62	2	0	6.8714		
5	Fort Lauderda	Florida	33311	South	FUR-TA-1000	Furniture	Tables	Bretford CR4	957.5775	5	0.45	-383.031		
6	Fort Lauderda	Florida	33311	South	OFF-ST-1000	Office Suppli	Storage	Eldon Fold 'N	22.368	2	0.2	2.5164		
7	Los Angeles	California	90032	West	FUR-FU-1000	Furniture	Furnishings	Eldon Expres	48.86	7	0	14.1694		
8	Los Angeles	California	90032	West	OFF-AR-1000	Office Suppli	Art	Newell 322	7.28	4	0	1.9656		
9	Los Angeles	California	90032	West	TEC-PH-1000	Technology	Phones	Mitel 5320 IP	907.152	6	0.2	90.7152		
10	Los Angeles	California	90032	West	OFF-BI-10003	Office Suppli	Binders	DXL Angle-Vi	18.504	3	0.2	5.7825		
11	Los Angeles	California	90032	West	OFF-AP-1000	Office Suppli	Appliances	Belkin F5C20	114.9	5	0	34.47		
12	Los Angeles	California	90032	West	FUR-TA-1000	Furniture	Tables	Chromcraft R	1706.184	9	0.2	85.3092		
13	Los Angeles	California	90032	West	TEC-PH-1000	Technology	Phones	Konftel 250 C	911.424	4	0.2	68.3568		
14	Concord	North Carolin	28027	South	OFF-PA-1000	Office Suppli	Paper	Xerox 1967	15.552	3	0.2	5.4432		
15	Seattle	Washington	98103	West	OFF-BI-10003	Office Suppli	Binders	Fellowes PB2	407.976	3	0.2	132.5922		
16	Fort Worth	Texas	76106	Central	OFF-AP-1000	Office Suppli	Appliances	Holmes Repl	68.81	5	0.8	-123.858		
17	Fort Worth	Texas	76106	Central	OFF-BI-1000	Office Suppli	Binders	Storex DuraT	2.544	3	0.8	-3.816		
18	Madison	Wisconsin	53711	Central	OFF-ST-1000	Office Suppli	Storage	Stur-D-Stor Sł	665.88	6	0	13.3176		
19	West Jordan	Utah	84084	West	OFF-ST-1000	Office Suppli	Storage	Fellowes Sup	55.5	2	0	9.99		
20	San Francisco	California	94109	West	OFF-AR-1000	Office Suppli	Art	Newell 341	8.56	2	0	2.4824		
21	San Francisco	California	94109	West	TEC-PH-1000	Technology	Phones	Cisco SPA 50	213.48	3	0.2	16.011		
22	San Francisco	California	94109	West	OFF-BI-10002	Office Suppli	Binders	Wilson Jones	22.72	4	0.2	7.384		
23	Fremont	Nebraska	68025	Central	OFF-AR-1000	Office Suppli	Art	Newell 318	19.46	7	0	5.0596		
24	Fremont	Nebraska	68025	Central	OFF-AP-1000	Office Suppli	Appliances	Acco Six-Outl	60.34	7	0	15.6884		

iv) In the similar way you can apply the other options such as Greater Than, Less that , Equal to , between etc.,

v) You can also select top "n" values for sales or profit by applying Top/ Bottom rules from conditional formatting.

ii) Color Scale Formatting Example

Color scale formatting Highlight the Sales values of each PRODUCT with **Color scale** conditional formatting.

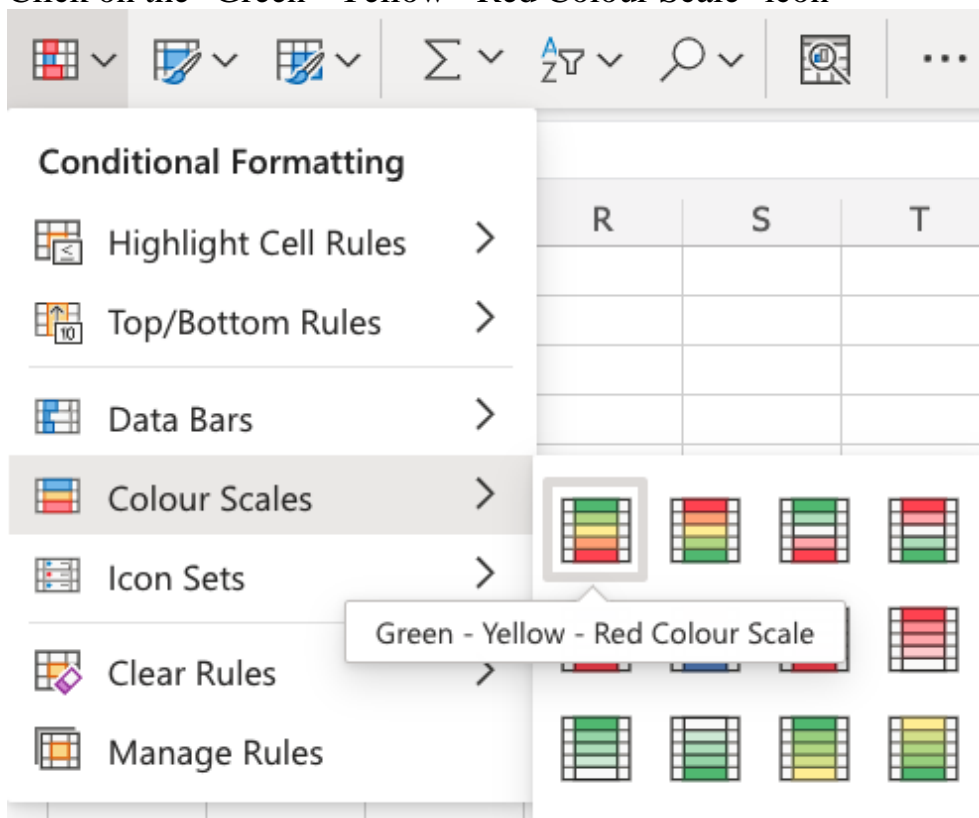Color Scale Conditional formatting, on sample super store as follows:

1. Select the range of sales values R2:end

2. Click on the Conditional Formatting icon in the ribbon, from the **Home** menu

3. Select the **Color Scales** from the drop-down menu
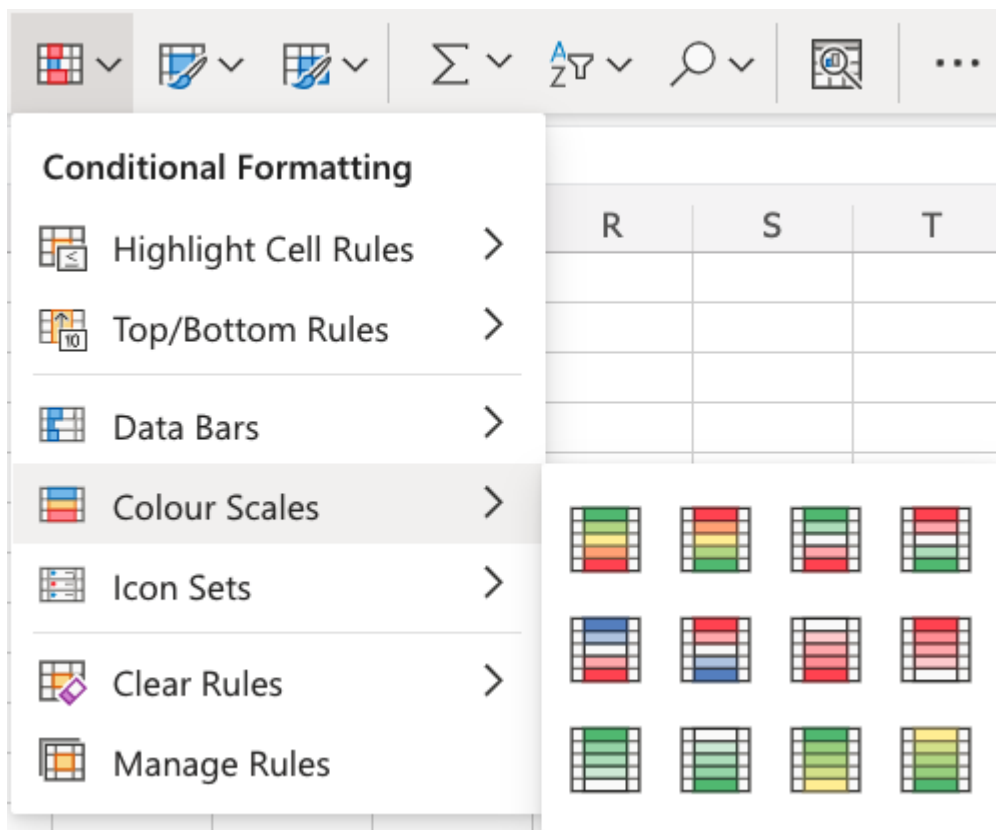
There are 12 Color Scale options with different color variations.

The color on the top of the icon will apply to the highest values.

4. Click on the "Green - Yellow - Red Colour Scale" icon

Now, the sales value cells will have a colored background highlighting:

Orange is used for the highest values, and dark red for the lowest values.
All the cells in the range gradually change color from yellow, orange, pink , dark red etc.

b) IF Function

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

Syntax:
=IF(**logical_test, [value_if_true], [value_if_false]**)

The **condition** is referred to as logical_test, 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 =

IF on Sample Super Store Data Set:
Create a new column called as discount status and enter following formula there
Implement if condition on Discount column as follows
   =IF(T2>0, "Discount allowed ", "Discount not allowed")

c) COUNTIF Function

Explanation:
The **COUNTIF** function is a premade function in Excel, which counts cells as specified.
Numbers (e.g. 90) and words (e.g. "Water") can be specified.
Syntax:
=COUNTIF(Range, criteria)

Where range = set of data on which count function is to be implemented

Criteria = condition that needs to be matched

On Sample Super Store Data Set,

Count the number of product that have sales more than 50 units as follows:
=countif(R2: End, ">50")

d) SUMIF Function

Explanation:

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**.

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.

Note: The [sum_range] is optional.
If not specified, the function calculates the sum of the same range as the condition.

On Sample Super Store Data Set

Count the total of profit that has profit more than 1000 units as follows:
=sumif(V2: End, ">1000")

e) AVERAGE Function

Explanation:

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

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

Note: The **AVERAGE** function ignores cells with text.

Syntax:

AVERAGE(number1, [number2], ...)

The AVERAGE function syntax has the following arguments:

- **Number1**   Required. The first number, cell reference, or range for which you want the average.
- **Number2, ...**   Optional. Additional numbers, cell references or ranges for which you want the average, up to a maximum of 255.

On Sample Super Store Data Set:

Find the average profit using following
= Average(R2:VEnd)
Where R2:REnd = cell address for sales

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

Syntax:

CONCAT(text1, [text2],…)

Argument	Description
text1 (required)	Text item to be joined. A string, or array of strings, such as a range of cells.
[text2, ...] (optional)	Additional text items to be joined. There can be a maximum of 253 text arguments for the text items. Each can be a string, or array of strings, such as a range of cells.

For example, =CONCAT("The"," ","sun"," ","will"," ","come"," ","up"," ","tomorrow.") will return **The sun will come up tomorrow.**

On the Sample Data Set:

Create a new column, label it as "Product Summary" and concat Product name, Sales and Profit for each product in that column as follows:
= concat(Q2,R2,V2)

2. INDEX, MATCH, UNIQUE , IFS, COUNTIFS, SUMIFS, AVERAGEIFS

a) Index

Explanation:

The Microsoft Excel INDEX function returns a value in a table based on the intersection of a row and column position within that table. The first row in the table is row 1 and the first column in the table is column 1.

The INDEX function is a built-in function in Excel that is categorized as a *Lookup/Reference Function*. It can be used as a worksheet function (WS) in Excel. As a worksheet function, the INDEX function can be entered as part of a formula in a cell of a worksheet.

Syntax:

The syntax for the INDEX function in Microsoft Excel is:

INDEX(table, row_number, column_number)

Parameters or Arguments

Table: A range of cells that contains the table of data.

row_number: The row position in the table where the value you want to lookup is located. This is the relative row position in the table and not the actual row number in the worksheet.

column_number: The column position in the table where the value you want to lookup is located. This is the relative column position in the table and not the actual column number in the worksheet.

Returns

The INDEX function returns any datatype such as a string, numeric, date, etc.

On sample Super data

set index function can be used to find the product

create two columns, one labelled as **Index value** and second labelled as **Row number** and enter the following formula in the column labelled index value as follows
=INDEX(O2:R20,3,2)   where O2:R20 = Cell addresses from category to sales

O/P= Binders

b) Match

Explanation:

The **MATCH** function searches for a specified item in a range of cells, and then returns the relative position of that item in the range.

Syntax:

MATCH(lookup_value, lookup_list, [match_type])

The MATCH function syntax has the following arguments:

- **lookup_value** Required. The value that you want to match in *lookup_array*. For example, when you look up someone's number in a telephone book, you are using the person's name as the lookup value, but the telephone number is the value you want. The *lookup_value* argument can be a value (number, text, or logical value) or a cell reference to a number, text, or logical value.
- **lookup_list** Required. The range of cells being searched.
- **match_type** Optional. The number -1, 0, or 1. The *match_type* argument specifies how Excel matches *lookup_value* with values in *lookup_array*. The default value for this argument is 1.
- The following table describes how the function finds values based on the setting of the *match_type* argument.

Match_type	Behavior
1 or omitted	**MATCH** finds the largest value that is less than or equal to *lookup_value*. The values in the *lookup_array* argument must be placed in ascending order, for example: ...-2, -1, 0, 1, 2, ..., A-Z, FALSE, TRUE.
0	**MATCH** finds the first value that is exactly equal to *lookup_value*. The values in the *lookup_array* argument can be in any order.
-1	**MATCH** finds the smallest value that is greater than or equal to*lookup_value*. The values in the *lookup_array* argument must be placed in descending order, for example: TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.

On Sample Super store:
Find the match of any product from the product name list as follows
= match("Xerox 1967", Q2 : End)

c) **Unique**
Explanation:
The UNIQUE function returns a list of unique values in a list or range.

Syntax:
=UNIQUE(array,[by_col],[exactly_once])
The UNIQUE function has the following arguments:

Argument	Description
Array Required	The range or array from which to return unique rows or columns

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

On Sample super store

We can find unique product categories by applying unique function on category column as follows

=unique(O2 : end)

Note: Unique function exists in 2016 onwards)

d) IFS

Explanation:

The IFS function checks whether one or more conditions are met, and returns a value that corresponds to the first TRUE condition. IFS can take the place of multiple nested IF statements, and is much easier to read with multiple conditions.

Syntax:

=IFS(logical_test1,   value_if_true1,   [logical_test2,   value_if_true2],   [logical_test3, value_if_true3],…)

Argument	Description
logical_test1 (required)	Condition that evaluates to TRUE or FALSE.
value_if_true1 (required)	Result to be returned if logical_test1 evaluates to TRUE. Can be empty.
logical_test2…logical_test127 (optional)	Condition that evaluates to TRUE or FALSE.
value_if_true2…value_if_true127 (optional)	Result to be returned if **logical_testN** evaluates to TRUE. Each **value_if_trueN** corresponds

with a condition **logical_testN**. Can be empty.

Example:

The formula in cell G7 is:

=IFS(F2=1,D2,F2=2,D3,F2=3,D4,F2=4,D5,F2=5,D6,F2=6,D7,F2=7,D8)

e) **COUNTIFS**
   Explanation:
   The **COUNTIFS** function applies criteria to cells across multiple ranges and counts the number of times all criteria are met.
   COUNTIFS(criteria_range1, criteria1, [criteria_range2, criteria2]…)
   The COUNTIFS function syntax has the following arguments:
   ▪ **criteria_range1**   Required. The first range in which to evaluate the associated criteria.
   ▪ **criteria1** Required. The criteria in the form of a number, expression, cell reference, or text that define which cells will be counted. For example, criteria can be expressed as 32, ">32", B4, "apples", or "32".
   ▪ **criteria_range2, criteria2, ...** Optional. Additional ranges and their associated criteria. Up to 127 range/criteria pairs are allowed.

   Syntax:

   COUNTIFS(criteria_range1, criteria1, [criteria_range2, criteria2]…)

   The COUNTIFS function syntax has the following arguments:

- **criteria_range1**   Required. The first range in which to evaluate the associated criteria.
- **criteria1**   Required. The criteria in the form of a number, expression, cell reference, or text that define which cells will be counted. For example, criteria can be expressed as 32, ">32", B4, "apples", or "32".
- **criteria_range2, criteria2, ...**   Optional. Additional ranges and their associated criteria. Up to 127 range/criteria pairs are allowed.

Example:

Data	
1	5/1/2011
2	5/2/2011
3	5/3/2011
4	5/4/2011
5	5/5/2011
6	5/6/2011

=COUNTIFS(A2:A7,"<6",A2:A7,">1")	Counts how many numbers between 1 and 6 (not including 1 and 6) are contained in cells A2 through A7.

f) SUMIFS
Explanation:
The SUMIFS function, one of the math and trig functions, adds all of its arguments that meet multiple criteria.
Syntax:

SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)

Argument name	Description
Sum_range (required)	The range of cells to sum.
Criteria_range1 (required)	The range that is tested using *Criteria1*. *Criteria_range1* and *Criteria1* set up a search pair whereby a range is searched for specific criteria. Once items in the range are found, their corresponding values in *Sum_range* are added.

Argument name		Description
Criteria1 (required)		The criteria that defines which cells in *Criteria_range1* will be added. For example, criteria can be entered as **32**, **">32"**, **B4**, **"apples"**, or **"32"**.
Criteria_range2, **criteria2,** **…** (optional)		Additional ranges and their associated criteria. You can enter up to 127 range/criteria pairs.

Example:

Quantity Sold	Product	Salesperson
5	Apples	Tom
4	Apples	Sarah
15	Artichokes	Tom
3	Artichokes	Sarah
22	Bananas	Tom
12	Bananas	Sarah
10	Carrots	Tom
33	Carrots	Sarah

=SUMIFS(A2:A9, B2:B9, "=A*", C2:C9, "Tom")	Adds the number of products that begin with **A** and were sold by **Tom**. It uses the wildcard character * in *Criteria1*, **"=A*"** to look for matching product names in *Criteria_range1* B2:B9, and looks for the name **"Tom"** in *Criteria_range2* C2:C9. It then adds the numbers in *Sum_range* A2:A9 that meet both conditions. The result is 20.

g) AVERAGEIFS

Explanation:
Returns the average (arithmetic mean) of all cells that meet multiple criteria.

Syntax:
AVERAGEIFS(average_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
The AVERAGEIFS function syntax has the following arguments:

- **Average_range** Required. One or more cells to average, including numbers or names, arrays, or references that contain numbers.
- **Criteria_range1, criteria_range2, …** Criteria_range1 is required, subsequent criteria_ranges are optional. 1 to 127 ranges in which to evaluate the associated criteria.
- **Criteria1, criteria2, ...** Criteria1 is required, subsequent criteria are optional. 1 to 127 criteria in the form of a number, expression, cell reference, or text that define which cells will be averaged. For example, criteria can be expressed as 32, "32", ">32", "apples", or B4.

Example:

Student	First	Second	Final
	Quiz	Quiz	Exam
	Grade	Grade	Grade
Emilio	75	85	87
Julie	94	80	88
Hans	86	93	Incomplete
Frederique	Incomplete	75	75

=AVERAGEIFS(B2:B5, B2:B5, ">70", B2:B5, "<90")	Average first quiz grade that falls between 70 and 90 for all students (80.5). The score marked "Incomplete" is not included in the calculation because it is not a numerical value.	O/P=75

3. VLOOKUP, HLOOKUP, XLOOKUP, COUNT, COUNTA

a) **VLOOKUP**
Explanation:
VLOOKUP is uses to find things in a table or a range by row. For example, look up a price of an automotive part by the part number, or find an employee name based on their employee ID.

Syntax :

=VLOOKUP (lookup_value, table_array, col_index_num, [range_lookup])

Argument name	Description

Argument name	Description
lookup_value (required)	The value you want to look up. The value you want to look up must be in the first column of the range of cells you specify in the *table_array* argument. For example, if *table-array* spans cells B2:D7, then your lookup_value must be in column B. *Lookup_value* can be a value or a reference to a cell.
table_array (required)	The range of cells in which the VLOOKUP will search for the *lookup_value* and the return value. You can use a named range or a table, and you can use names in the argument instead of cell references. The first column in the cell range must contain the *lookup_value*. The cell range also needs to include the return value you want to find. Learn how to select ranges in a worksheet.
col_index_num (required)	The column number (starting with 1 for the left-most column of *table_array*) that contains the return value.
range_lookup (optional)	A logical value that specifies whether you want **VLOOKUP** to find an approximate or an exact match: ▪ **Approximate match - 1/TRUE** assumes the first column in the table is sorted either numerically or alphabetically, and will then search for the closest value. This is the default method if you don't specify one. For example, =VLOOKUP(90,A1:B100,2,TRUE). ▪ **Exact match - 0/FALSE** searches for the exact value in the first column. For example, =VLOOKUP("Smith",A1:B100,2,FALSE).

On sample super store:
We can lookup for product name and display its profit as follows:

=VLOOKUP("Product name', range of cells, 3, 0)

b) HLOOKUP

Explanation:

Searches for a value in the top row of a table or an array of values, and then returns a value in the same column from a row you specify in the table or array. Use HLOOKUP when your comparison values are located in a row across the top of a table of data, and you want to look down a specified number of rows. Use VLOOKUP when your comparison values are located in a column to the left of the data you want to find.

The H in HLOOKUP stands for "Horizontal."

Syntax:

=HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])

The HLOOKUP function syntax has the following arguments:

- **Lookup_value**   Required. The value to be found in the first row of the table. Lookup_value can be a value, a reference, or a text string.
- **Table_array**   Required. A table of information in which data is looked up. Use a reference to a range or a range name.
- The values in the first row of table_array can be text, numbers, or logical values.
- If range_lookup is TRUE, the values in the first row of table_array must be placed in ascending order: ...-2, -1, 0, 1, 2,... , A-Z, FALSE, TRUE; otherwise, HLOOKUP may not give the correct value. If range_lookup is FALSE, table_array does not need to be sorted.
- Uppercase and lowercase text are equivalent.
- Sort the values in ascending order, left to right. For more information, see Sort data in a range or table.
- **Row_index_num** Required. The row number in table_array from which the matching value will be returned. A row_index_num of 1 returns the first row value in table_array, a row_index_num of 2 returns the second row value in table_array, and so on. If row_index_num is less than 1, HLOOKUP returns the #VALUE! error value; if row_index_num is greater than the number of rows on table_array, HLOOKUP returns the #REF! error value.
- **Range_lookup**   Optional. A logical value that specifies whether you want HLOOKUP to find an exact match or an approximate match. If TRUE or omitted, an approximate match is returned. In other words, if an exact match is not found, the next largest value that is less than lookup_value is returned. If FALSE, HLOOKUP will find an exact match. If one is not found, the error value #N/A is returned.

On sample super store:

We can lookup for product name and display its profit as follows:

=VLOOKUP("Product name', range of cells, 3, 0)

c) XLOOKUP

Explanation:

XLOOKUP function is used to find things in a table or range by row. For example, look up the price of an automotive part by the part number, or find an employee name based on their employee ID. With XLOOKUP, you can look in one column for a search term and return a result from the same row in another column, regardless of which side the return column is on.

The XLOOKUP function searches a range or an array, and then returns the item corresponding to the first match it finds. If no match exists, then XLOOKUP can return the closest (approximate) match.

Syntax:

=XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])

Argument	Description
lookup_value Required*	The value to search for *If omitted, XLOOKUP returns blank cells it finds in **lookup_array**.
lookup_array Required	The array or range to search
return_array Required	The array or range to return
[if_not_found] Optional	Where a valid match is not found, return the [if_not_found] text you supply. If a valid match is not found, and [if_not_found] is missing, **#N/A** is returned.
[match_mode] Optional	Specify the match type: 0 - Exact match. If none found, return #N/A. This is the default. -1 - Exact match. If none found, return the next smaller item. 1 - Exact match. If none found, return the next larger item. 2 - A wildcard match where *, ?, and ~ have special meaning.
[search_mode] Optional	Specify the search mode to use: 1 - Perform a search starting at the first item. This is the default. -1 - Perform a reverse search starting at the last item.

Argument	Description
	2 - Perform a binary search that relies on lookup_array being sorted in *ascending* order. If not sorted, invalid results will be returned. -2 - Perform a binary search that relies on lookup_array being sorted in *descending* order. If not sorted, invalid results will be returned.

d) COUNT

Explanation:

The **COUNT** function counts the number of cells that contain numbers, and counts numbers within the list of arguments. Use the **COUNT** function to get the number of entries in a number field that is in a range or array of numbers.

Syntax:

COUNT(value1, [value2], ...)

The COUNT function syntax has the following arguments:

▪ **value1**   Required. The first item, cell reference, or range within which you want to count numbers.

▪ **value2, ...**   Optional. Up to 255 additional items, cell references, or ranges within which you want to count numbers.

Note: The arguments can contain or refer to a variety of different types of data, but only numbers are counted.

On Sample Super Store:

=COUNT(R2:Rend)   where R2: Rend = address of sales column

e) COUNTA

Explanation:

The **COUNTA** function counts the number of cells that are not empty in a range.

Syntax:

COUNTA(value1, [value2], ...)

The COUNTA function syntax has the following arguments:

▪ **value1**   Required. The first argument representing the values that you want to count.

▪ **value2, ...**   Optional. Additional arguments representing the values that you want to count, up to a maximum of 255 arguments.

Remarks

▪ The **COUNTA** function counts cells containing any type of information, including error values and empty text (**""**). For example, if the range contains a formula that returns an empty string, the **COUNTA** function counts that value.

The **COUNTA** function does not count empty cells.

- If you do not need to count logical values, text, or error values (in other words, if you want to count only cells that contain numbers), use the **COUNT** function.
- If you want to count only cells that meet certain criteria, use the **COUNTIF** function or the **COUNTIFS** function

On Sample super store:

=COUNTA(Q2:Qend)　　　　　where Q2: QEnd = column address of Product name

4. LEFT, MID, RIGHT, LEN, SUBSTITUTE, SEARCH, ISNUMBER

a) LEFT
Explanation:

LEFT returns the first character or characters in a text string, based on the number of characters you specify.

Syntax:

=**LEFT**(**text**,**num_chars**)

Text is the text string that contains the characters you want to extract.

Num_chars specifies the number of characters you want LEFT to extract. Num_chars must be greater than or equal to zero. If num_chars is greater than the length of text, LEFT returns all of text. If num_chars is omitted, it is assumed to be till the end of the string.

Example:

= left ("Data Analytics", 3)

O/P = "Dat"

b) MID

Explanation:

MID returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.

Syntax:

=**MID**(**text**,**start_num**,**num_chars**)

Text is the text string containing the characters you want to extract.

Start_num is the position of the first character you want to extract in text. The first character in text has start_num 1, and so on.

Num_chars specifies the number of characters you want MID to return from text.

Example:

= mid ("Data Analytics", 3, 4)

O/P = "ta A"

c) RIGHT

Explanation:
RIGHT returns the last character or characters in a text string, based on the number of characters you specify.

Syntax
=RIGHT(text,[num_chars])

- **Text**   Required. The text string containing the characters you want to extract.
- **Num_chars**   Optional. Specifies the number of characters you want RIGHT to extract.
- Num_chars must be greater than or equal to zero.
- If num_chars is greater than the length of text, RIGHT returns all of text.
- If num_chars is omitted, it is assumed to be 1.
  Example:
  = right ("Data Analytics", 3)
  O/P = "ics"

d) LEN
Explanation:
LEN returns the number of characters in a text string.

Syntax:
=LEN(text)
The LEN function syntax has the following arguments:
- **Text**   Required. The text whose length you want to find. Spaces count as characters.
  Example:
  = len("Data Analytics")
  O/P = 14

e) SUBSTITUTE
Explanation:
Substitutes new_text for old_text in a text string. Use SUBSTITUTE when you want to replace specific text in a text string; use REPLACE when you want to replace any text that occurs in a specific location in a text string.
Syntax:

=SUBSTITUTE(text, old_text, new_text, [instance_num])
The SUBSTITUTE function syntax has the following arguments:
- **Text**   Required. The text or the reference to a cell containing text for which you want to substitute characters.

- **Old_text**   Required. The text you want to replace.
- **New_text**   Required. The text you want to replace old_text with.
- **Instance_num**   Optional. Specifies which occurrence of old_text you want to replace with new_text. If you specify instance_num, only that instance of old_text is replaced. Otherwise, every occurrence of old_text in text is changed to new_text.

Example:
        =SUBSTITUTE("Data Analytics", "Data", "Big Data")

O/P = Big Data Analytics

f) SEARCH

Explanation:

The **SEARCH** functions locate one text string within a second text string, and return the number of the starting position of the first text string from the first character of the second text string. For example, to find the position of the letter "n" in the word "printer", you can use the following function:

=SEARCH("n","printer")

This function returns **4** because "n" is the fourth character in the word "printer."
You can also search for words within other words. For example, the function

=SEARCH("base","database")

returns **5**, because the word "base" begins at the fifth character of the word "database". You can use

Syntax:

  = SEARCH(find_text,within_text,[start_num])

- **find_text**   Required. The text that you want to find.
- **within_text**   Required. The text in which you want to search for the value of the *find_text* argument.
- **start_num**   Optional. The character number in the *within_text* argument at which you want to start searching.

On Sample Super store

    =SEARCH("Supplies",O4)
    Where Q4 = cell address of category column

g) ISNUMBER

Explanation:

    The ISNUMBER function returns TRUE when a cell contains a number, and FALSE if not. You can use ISNUMBER to check that a cell contains a numeric value, or that the result of another function is a number.

Syntax

=ISNUMBER(value)

Arguments

value - The value to check.

Example:

   (1)=ISNUMBER(9.8)   O/P= True

   (2)=ISNUMBER("AA")   O/P= False

5. TODAY, NOW, YEAR, MONTH, NETWORKDAYS, EOMONTH

Note: for this problem we don't need sample super store data set

a) TODAY

Returns the current date

Syntax:

=today()

b) Now

Returns current date and time

Syntax:

=now()

c) Year

Returns the year corresponding to a date. The year is returned as an integer in the range 1900-9999.

Syntax

YEAR(serial_number)

The YEAR function syntax has the following arguments:

- **Serial_number**   Required. The date of the year you want to find. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates are entered as text.

Example:

=year(today())

d) Month

Returns the month of a date represented by a serial number. The month is given as an integer, ranging from 1 (January) to 12 (December).

Syntax

MONTH(serial_number)

The MONTH function syntax has the following arguments:

- **Serial_number**   Required. The date of the month you are trying to find. Dates should be entered by using the DATE function, or as results of other formulas or functions.

e) Networkdays

Returns the number of whole working days between start_date and end_date. Working days exclude weekends and any dates identified in holidays.

Syntax
NETWORKDAYS(start_date, end_date, [holidays])
The NETWORKDAYS function syntax has the following arguments:
- **Start_date**   Required. A date that represents the start date.
- **End_date**   Required. A date that represents the end date.
- **Holidays**   Optional. An optional range of one or more dates to exclude from the working calendar, such as state and federal holidays and floating holidays. The list can be either a range of cells that contains the dates or an array constant of the serial numbers that represent the dates.

Example
Copy the example data in the following table, and paste it in cell A1 of a new Excel worksheet. For formulas to show results, select them, press F2, and then press Enter. If you need to, you can adjust the column widths to see all the data.

Date	Description	
10/1/2012	Start date of project	
3/1/2013	End date of project	
11/22/2012	Holiday	
12/4/2012	Holiday	
1/21/2013	Holiday	
Formula	**Description**	**Result**
=NETWORKDAYS(A2,A3)	Number of workdays between the start (10/1/2012) and end date (3/1/2013).	110
=NETWORKDAYS(A2,A3,A4)	Number of workdays between the start (10/1/2012) and end date (3/1/2013), with the 11/22/2012 holiday as a non-working day.	109
=NETWORKDAYS(A2,A3,A4:A6)	Number of workdays between the start (10/1/2012) and end date (3/1/2013), with the three holidays as non-working days.	107

f)   Eomonth
Returns the serial number for the last day of the month that is the indicated number of months before or after start_date. Use EOMONTH to calculate maturity dates or due dates that fall on the last day of the month.

Syntax:

EOMONTH(start_date, months)

The EOMONTH function syntax has the following arguments:

- **Start_date**   Required. A date that represents the starting date. Dates should be entered by using the DATE function, or as results of other formulas or functions. For example, use DATE(2008,5,23) for the 23rd day of May, 2008. Problems can occur if dates are entered as text.
- **Months**   Required. The number of months before or after start_date. A positive value for months yields a future date; a negative value yields a past date.

  Note: If months is not an integer, it is truncated.

Example

Copy the example data in the following table, and paste it in cell A1 of a new Excel worksheet. For formulas to show results, select them, press F2, and then press Enter. If you need to, you can adjust the column widths to see all the data.

Date
1-Jan-11

Formula	Description	Result
=EOMONTH(A2,1)	Date of the last day of the month, one month after the date in A2.	2/28/2011
=EOMONTH(A2,-3)	Date of the last day of the month, three months before the date in A2.	10/31/2010

6. FILTER, FREQUENCY, SEQUENCE, RANDARRAY, IFERROR

a) FILTER

Explanation:

The **FILTER** function allows you to filter a range of data based on criteria you define.

Syntax:

=FILTER(array,include,[if_empty])

Argument	Description
Array	The array, or range to filter

Required

Include	A Boolean array whose height or width is the same as the array
Required	

[if_empty]	
Optional	

On Sample Super Store Data set:
 Filter the values for profit greater than 100 units as
= filter (U2:End, U2: UEnd > 100)

Where U2: UEnd = Cell address for profit

b) FREQUENCY

Explanation:
The FREQUENCY function calculates how often values occur within a range of values, and then returns a vertical array of numbers.
Syntax:

FREQUENCY(data_array, bins_array)

The FREQUENCY function syntax has the following arguments:

- **data_array**   Required. An array of or reference to a set of values for which you want to count frequencies. If data_array contains no values, FREQUENCY returns an array of zeros.

- **bins_array**   Required. An array of or reference to intervals into which you want to group the values in data_array. If bins_array contains no values, FREQUENCY returns the number of elements in data_array.

Example:
Use this Data Set for frequency:

Scores	Bins
79	70
85	79
78	89
85	

50	
81	
95	
88	
97	

Steps:

1) Create column Bins = upper limit of your class interval
2) Select cell array where you need formula. Type following formula in the formula bar and then press

SHIFT + CTRL + ENTER

Formula=frequency(A2:A10,B2:B4)

O/P-

interval	Frequency
<70	1
71-79	2
80-89	4
90>	2

c) SEQUENCE

Explanation:
The SEQUENCE function allows you to generate a list of sequential numbers in an array, such as 1, 2, 3, 4.

Syntax:
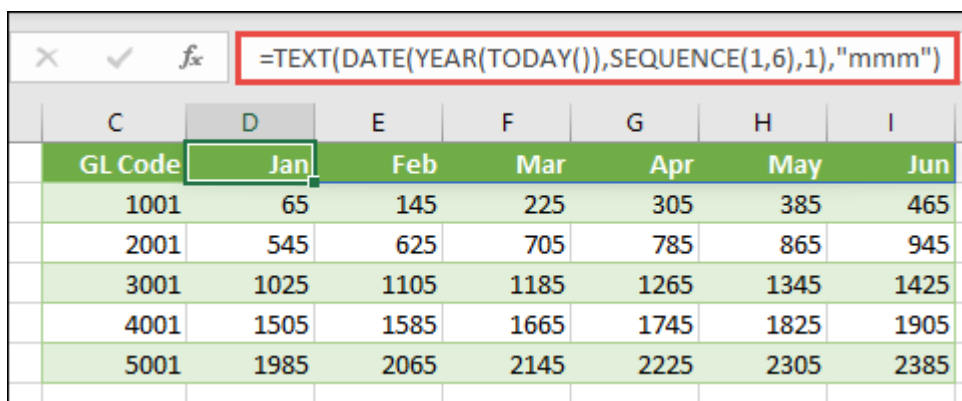
=SEQUENCE(rows,[columns],[start],[step])

Argument	Description
Rows Required	The number of rows to return
[columns] Optional	The number of columns to return
[start] Optional	The first number in the sequence

Argument	Description
[step] Optional	

Example:

If you need to create a quick sample dataset, here's an example using SEQUENCE with TEXT, DATE, YEAR, and TODAY to create a dynamic list of months for a header row, where the underlying date will always be the current year. Our formula is: **=TEXT(DATE(YEAR(TODAY()),SEQUENCE(1,6),1),"mmm")**.

	=TEXT(DATE(YEAR(TODAY()),SEQUENCE(1,6),1),"mmm")					
C	D	E	F	G	H	I
GL Code	Jan	Feb	Mar	Apr	May	Jun
1001	65	145	225	305	385	465
2001	545	625	705	785	865	945
3001	1025	1105	1185	1265	1345	1425
4001	1505	1585	1665	1745	1825	1905
5001	1985	2065	2145	2225	2305	2385

d) **RANDARRAY**

Explanation:
The **RANDARRAY** function returns an array of random numbers. You can specify the number of rows and columns to fill, minimum and maximum values, and whether to return whole numbers or decimal values.

Syntax:

=RANDARRAY([rows],[columns],[min],[max],[whole_number])

Argument	Description

Argument	Description
[rows]	The number of rows to be returned
Optional	
[columns]	The number of columns to be returned
Optional	
[min]	The minimum number you would like returned
Optional	
[max]	The maximum number you would like returned
Optional	
[whole_number]	Return a whole number or a decimal value
Optional	- TRUE for a whole number - FALSE for a decimal number.

Example:
1) In the example shown, RANDARRAY is used to generate 50 values in a range of 10 rows by 5 columns. The formula in B4 is:

   =RANDARRAY(10,5)

2) To return a random array of integers, 5 rows by 2 columns, between 1 and 10, you can use a formula like this:

   =RANDARRAY(5,2,1,10,TRUE)

e) IFERROR

Explanation:
You can use the IFERROR function to trap and handle errors in a formula. IFERROR returns a value you specify if a formula evaluates to an error; otherwise, it returns the result of the formula.

Syntax:

IFERROR(value, value_if_error)

The IFERROR function syntax has the following arguments:

- **value**   Required. The argument that is checked for an error.
  - **value_if_error**   Required. The value to return if the formula evaluates to an error. The following error types are evaluated: #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!.

Example:

Quota	Units Sold
210	35
55	0
	23

=IFERROR(A2/B2, "Error in calculation")	Checks for an error in the formula in the first argument (divide 210 by 35), finds no error, and then returns the results of the formula	O/P=6

7. PIVOT TABLES, WHAT IF ANALYSIS, DATA VALIDATION, SUBTOTALS WITH RANGES

a) PIVOT TABLES

Explanation:

A **PivotTable** is a powerful tool to calculate, summarize, and analyze data that lets you see comparisons, patterns, and trends in your data.

Steps:

1. Select the cells you want to create a PivotTable from.

2. Select **Insert** > **PivotTable**.
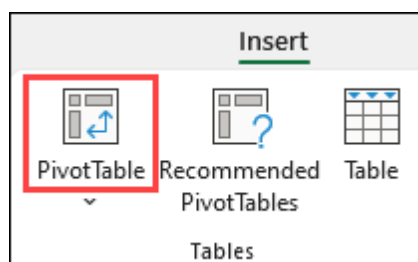
3. This creates a PivotTable based on an existing table or range.
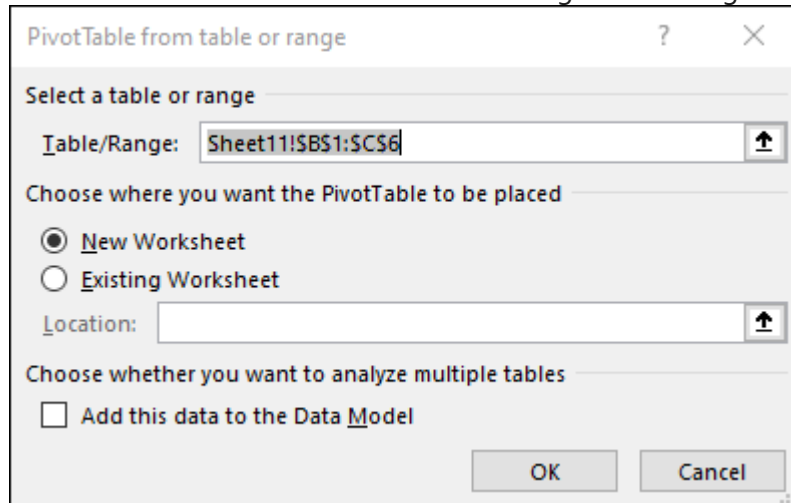
Note: Selecting **Add this data to the Data Model** adds the table or range being used for this PivotTable into the workbook's Data Model. Learn more.

4. Choose where you want the PivotTable report to be placed. Select **New Worksheet** to place the PivotTable in a new worksheet or **Existing Worksheet** and select where you want the new PivotTable to appear.

5. Select **OK**.

b) WHAT IF ANALYSIS
Explanation:
Excel What-if Analysis is a procedure employed to the Excel sheets with formulas to see the tabular data results when any variations are applied to the original values without having to recreate a new sheet. We have three types of What-if Analysis as shown below.

a) What-If Analysis Scenario Manager

Scenario Manager creates scenarios for each set of the input values for the variables under consideration. Scenarios help you to explore a set of possible outcomes,

Example:

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.
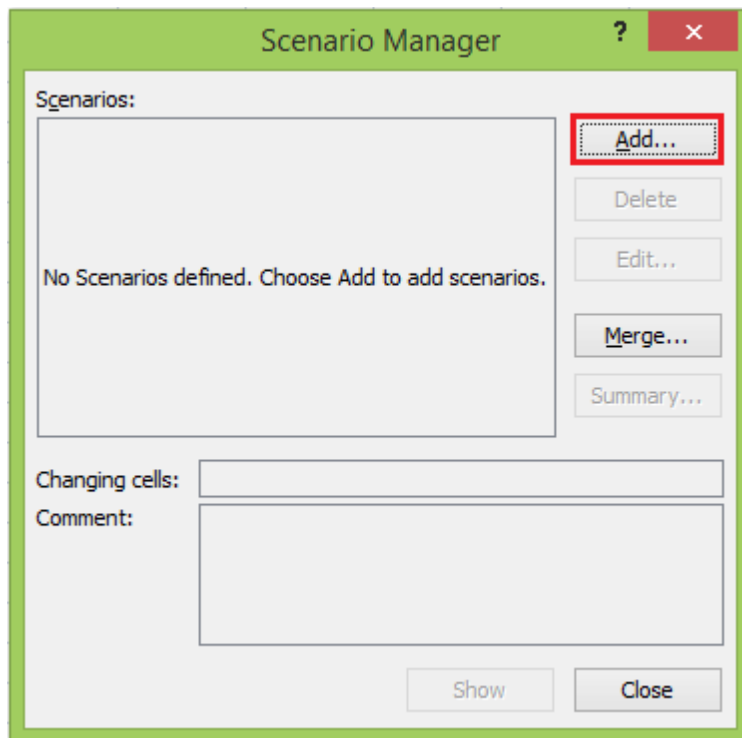
Step 6: Below is the created table.

To create an analysis report with Scenario Manager, follow the following steps, such as:
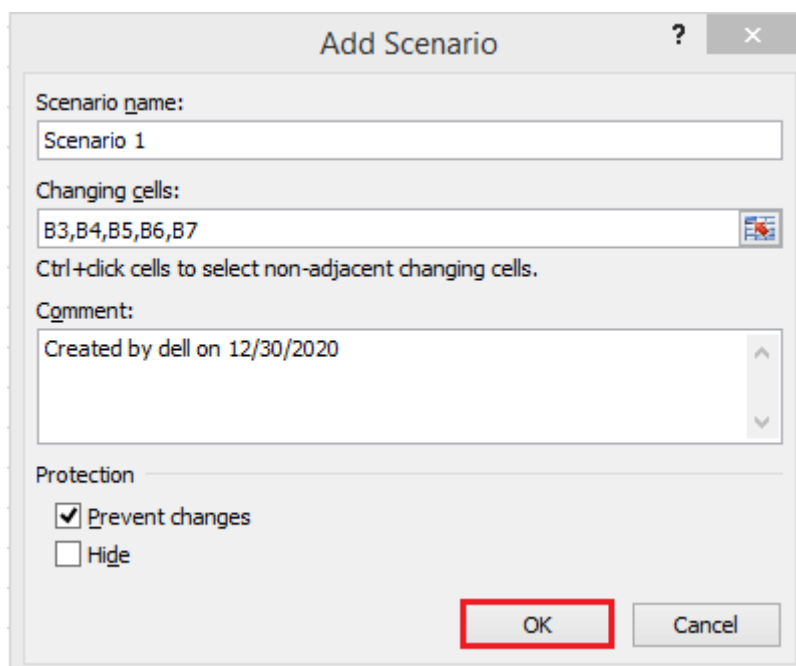
Step 1: Click the **Data** tab.

Step2: Go to the **What-If Analysis** button and click on the **Scenario Manager** from the dropdown list.
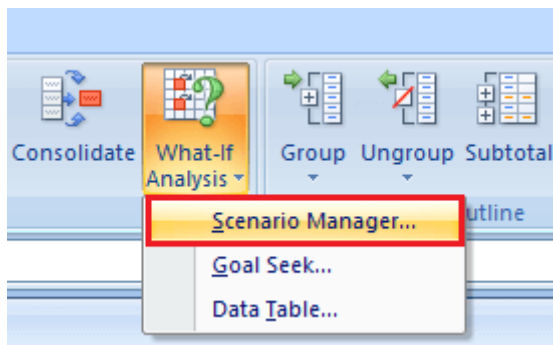
Step 3: Now a scenario manager dialog box appears, click on the **Add** button to create a scenario.

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.

o   In the scenario name box, create scenario 2.

o   Select the prevent changes.

o   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.
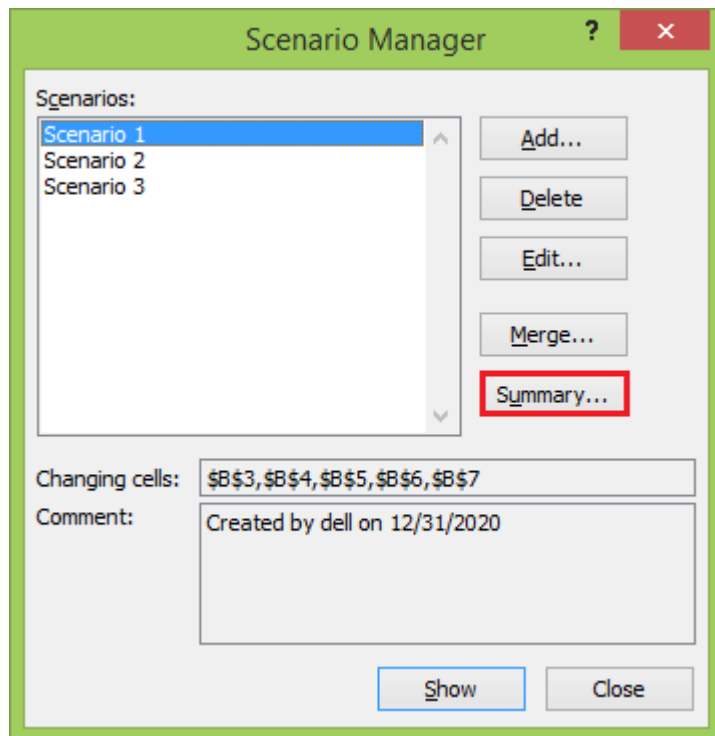
Step 13: Now, click on the **Summary** button. The Scenario Summary dialog box appears.
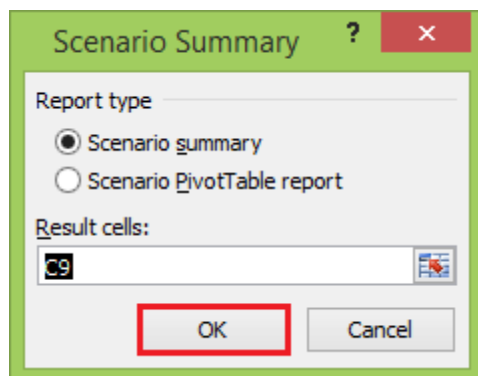
b) What-If Analysis Goal Seek

Explanation:

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.

Example: Below are the following steps to use the Goal Seek feature in Excel.

Data Set

Internal1 Marks	45
Internal2 Marks	34
Assignment Marks	50
Lab Marks	35
Improvement Test Marks?	36
Grade	200

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 C9 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 57000 in the **To value** box. Here, you get the formula result.

Step 5: Type B9 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**.

Step 7: Goal Seek box produces the following result.

As you can observe, Goal Seek found the solution using B9, and it returns 0 in the B9 cell because the target value and current value are the same.

c) What-If Analysis Data Tables

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:

Example:

1) Enter following data set Sales , unit price and month with data 500, 55, 11
2) Amount is calculated using formula in B6 : = sales * unitprice * month
$$= B2 * B3 * B4$$
3) In column D2 enter the following formula " = B6
4) Enter values 500, 600 , 700 so on in cell D3, D4, D5, D6 …….
5) Enter month values in rows 1,2,3,4, in F2, G2, H2, I2, …….
6) Select table array from D2 to Iend.
7) Click Data Tab → select What if analysis →select Data Table
8) Give Row input → B5 (Cell address of month)
   Give Column input → B3 (Cell address of sales)
9)Press Enter and you will get the Data table of the amount calculated for remaining months.

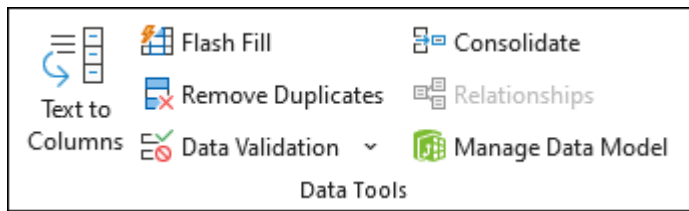|	A	B	C	D	E	F	G	H	I	J
1										
2	SALES	500								
3	UNIT PRICE	55								
4	MONTH	1								
5										
6	AMOUNT									
7										
8										
9										
10										
11										
12										

c) DATA VALIDATION
Explanation:

Data validation is used to restrict the type of data or the values that users enter into a cell, like a dropdown list.

Steps:

1. Select the cell(s) you want to create a rule for.
2. Select **Data >Data Validation**.

3. On the **Settings** tab, under **Allow**, select an option:
- **Whole Number** - to restrict the cell to accept only whole numbers.
- **Decimal** - to restrict the cell to accept only decimal numbers.
- **List** - to pick data from the drop-down list.
- **Date** - to restrict the cell to accept only date.
- **Time** - to restrict the cell to accept only time.
- **Text Length** - to restrict the length of the text.
- **Custom** – for custom formula.
4. Under **Data**, select a condition.
5. Set the other required values based on what you chose for **Allow** and **Data**.
6. Select the **Input Message** tab and customize a message users will see when entering data.
7. Select the **Show input message when cell is selected** checkbox to display the message when the user selects or hovers over the selected cell(s).
8. Select the **Error Alert** tab to customize the error message and to choose a **Style**.
9. Select **OK**.

Now, if the user tries to enter a value that is not valid, an **Error Alert** appears with your customized message.

d) SUBTOTALS WITH RANGES
Explanation:

You can automatically calculate subtotals and grand totals in a list for a column by using the **Subtotal** command.

Example: On Sample super store (Order sheet)
Steps

1) Sort the data in descending order using sort option in Data tab. Sort on region data.
2) Click on subtotal option from data tab→ select region in at each value change field
3) Use function → Select sum
4) Add subtotal to→ Select Sum
5) You will get subtotal region wise

8. DEVELOP AN INTERACTIVE DASHBOARD FOR THE FINANCIAL SAMPLE EXCEL WORKBOOK

Solution :

Step 1: Write the Problem Statements(Goals)

1) Category wise profit
2) Region wise profit
3) Segment wise Profit
   Doughnut chart
4) What are the total sales by segment in each / all years?
5) What is the total sales in different / all categories
6) What is the total sales in different / all region.
   Bar chart
7) What are the profit and sales in different regions
   Bar chart
8) What is total quantity in different / all categories
   Pie chart

Steps to create a Dashboard on Sample Super Store
1) Data cleaning:
a) Add filters to all the columns
b) For each column check the filters for any data anomaly such as blank cells, incorrect data etc.
c) If any anomalies correct them manually
2) For every Goal defined above do the analysis in the following manner:
a) Insert Pivot table in the new sheet and include all the columns from the data set.
b) Drag the field on which analysis has to be done into the respective tab of pivot table field.
c) Remove the grand total from the pivot data using following steps
   Select the pivot table data from the excel sheet and click on design ribbon → click grand total → Select off for rows and columns.
d) Select options ribbon → click on pivot chart option from the ribbon → select type of chart from the menu → click on ok
e) You will get a desired chart which can be formatted using various options from the pivot chart tools ribbon

3) Copy all the charts create by following step 2 into a new sheet. Name the sheet as Excel Dashboard
4) In this excel dashboard sheet
   go to pivot chart tools ribbon -→ select Analyze option → select insert slicer→select columns on which you need to insert slicer
5) Connect all the slicers using following steps
   Goto slicer tools→ click on options → select pivot table connections → check all the pivot tables you need to connect.
6) This dashboard will be interactive. When you select data from the slicer, your charts will dynamically update to display the selected options.
   O/P: Dashboard