Part- A: Spreadsheet (Excel)

Data preprocessing, interpretation and analytical functions

Note: Download the sample data file from the open sources (Kaggle.com, etc.,) to apply & practice all these functions.

1. CONDITIONAL FORMATTING, IF, COUNTIF, SUMIF, AVERAGE, CONCAT
2. INDEX, MATCH, UNIQUE, IFS, COUNTIFS, SUMIFS, AVERAGEIFS
3. VLOOKUP, HLOOKUP, XLOOKUP, COUNT, COUNTA
4. LEFT, MID, RIGHT, LEN, SUBSTITUTE, SEARCH, ISNUMBER
5. TODAY, NOW, YEAR, MONTH, NETWORKDAYS, EOMONTH
6. OFFSET, CHOOSE, LET, MAX, SORT, SORTBY, RANK
7. FILTER, FRQUENCY, SEQUENCE, RANDARRAY, IFERROR
8. PIVOT TABLES, WHAT-IF ANALYSIS, DATA VALIDATION, SUBTOTALS WITH RANGES
9. Develop an interactive dashboard for the Financial Sample Excel workbook (https://learn.microsoft.com/en-us/power-bi/create-reports/sample-financial-download) or Sample-Superstore Excel data

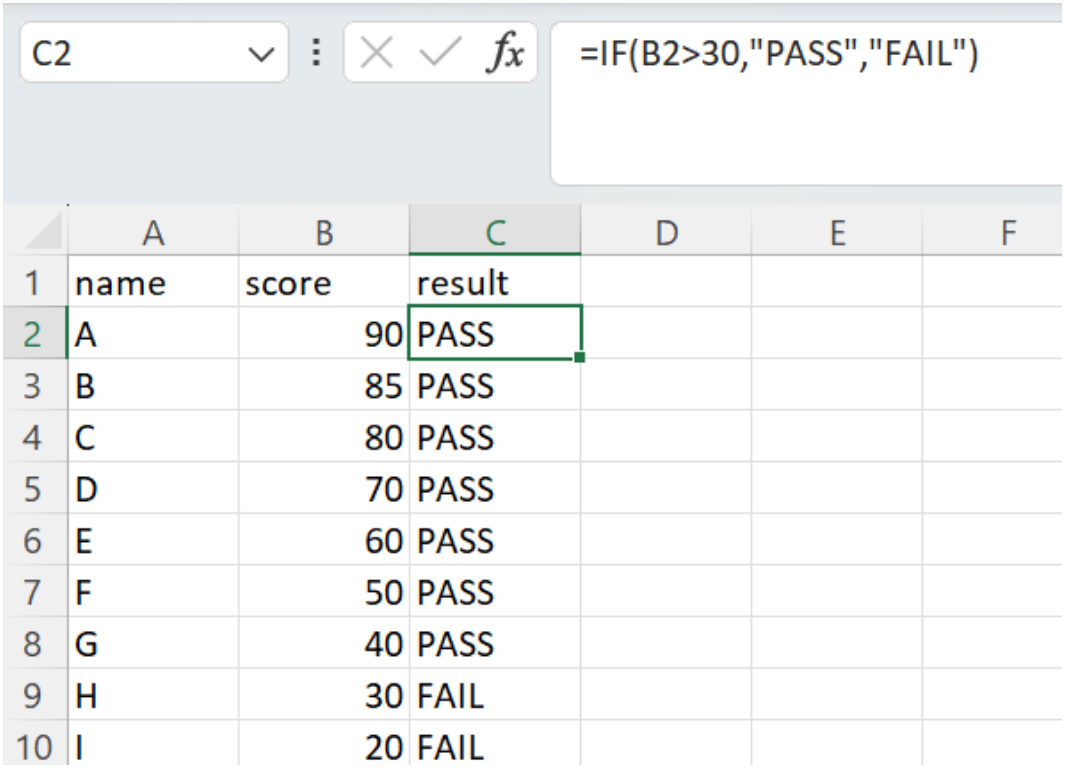
1. IF FUNCTION

The Excel IF function runs a logical test and returns one value for a TRUE result, and another for a FALSE result.

Syntax

```
=IF(logical_test,[value_if_true],[value_if_false])
```

- *logical_test* - A value or logical expression that can be evaluated as TRUE or FALSE.
- *value_if_true* - [optional] The value to return when logical_test evaluates to TRUE.
- *value_if_false* - [optional] The value to return when logical_test evaluates to FALSE.



	A	B	C	D	E	F
1	name	score	result			
2	A	90	PASS			
3	B	85	PASS			
4	C	80	PASS			
5	D	70	PASS			
6	E	60	PASS			
7	F	50	PASS			
8	G	40	PASS			
9	H	30	FAIL			
10	I	20	FAIL			

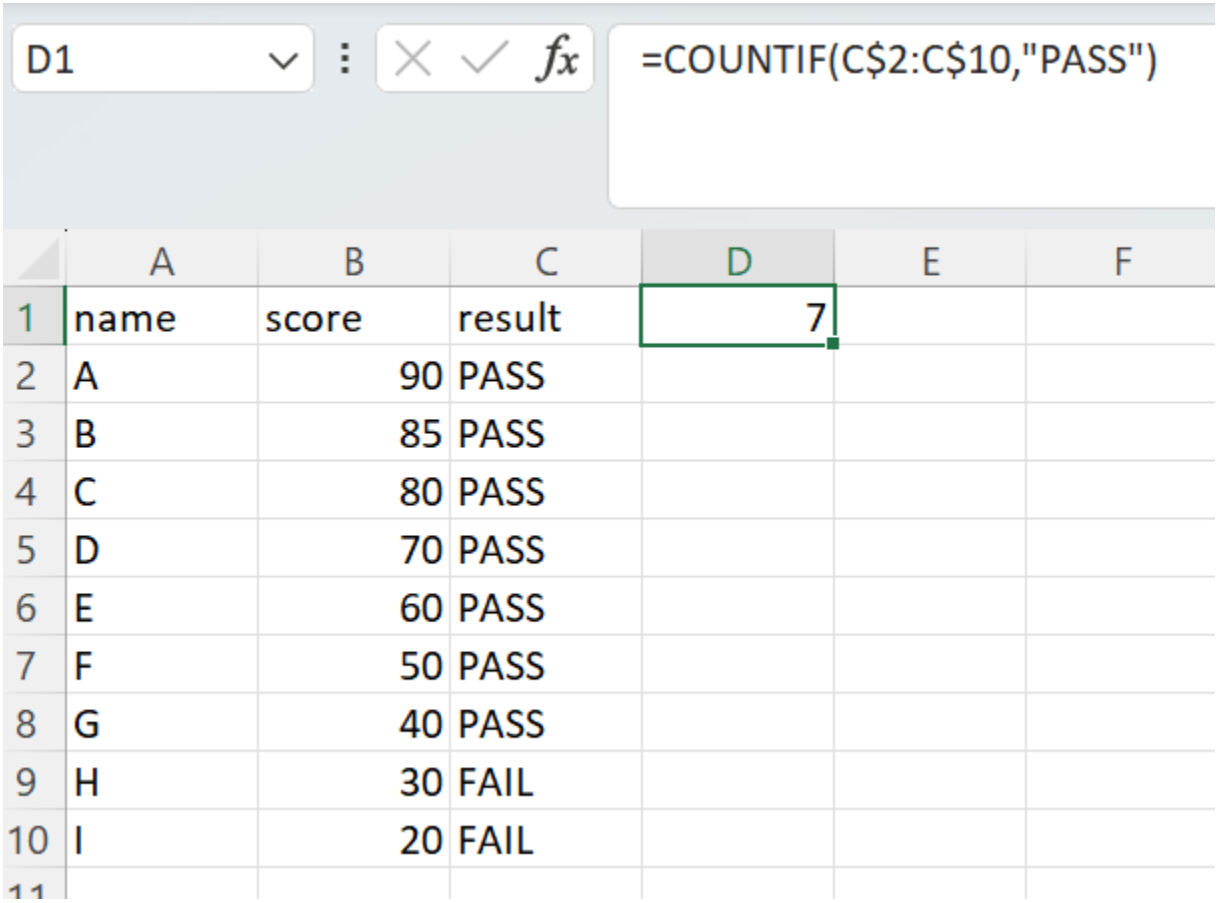
2. COUNTIF

The Excel COUNTIF function returns the count of cells in a range that meet a single condition.

Syntax

```
=COUNTIF(range,criteria)
```

- *range* - The range of cells to count.
- *criteria* - The criteria that controls which cells should be counted.



	A	B	C	D	E	F
1	name	score	result	7		
2	A	90	PASS			
3	B	85	PASS			
4	C	80	PASS			
5	D	70	PASS			
6	E	60	PASS			
7	F	50	PASS			
8	G	40	PASS			
9	H	30	FAIL			
10	I	20	FAIL			

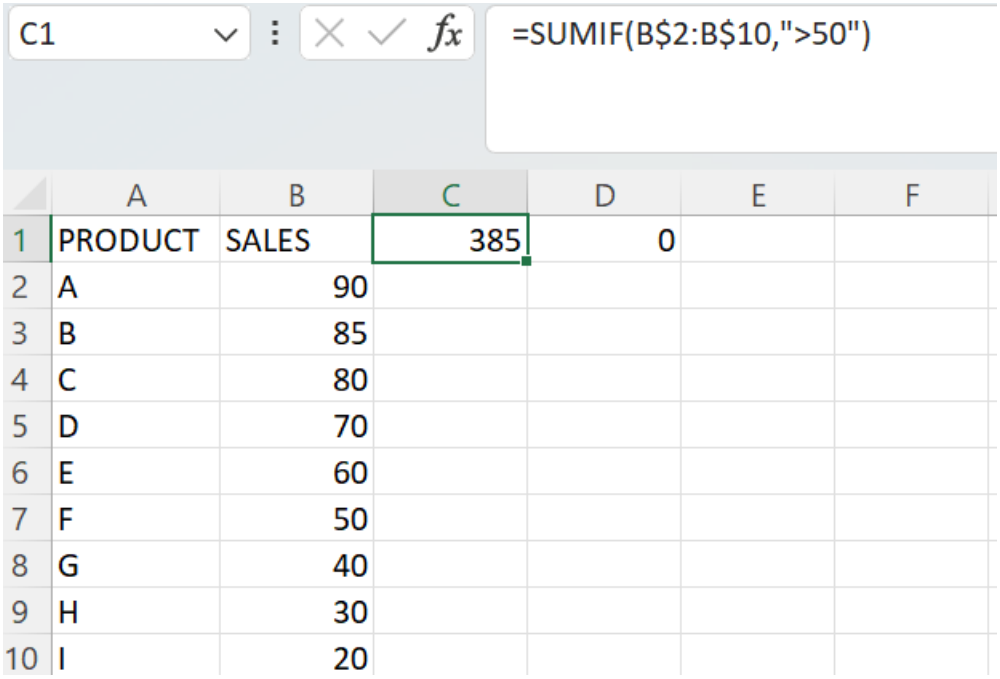
3. SUMIF

The Excel SUMIF function returns the sum of cells that meet a single condition.

Syntax

```
=SUMIF(range,criteria,[sum_range])
```

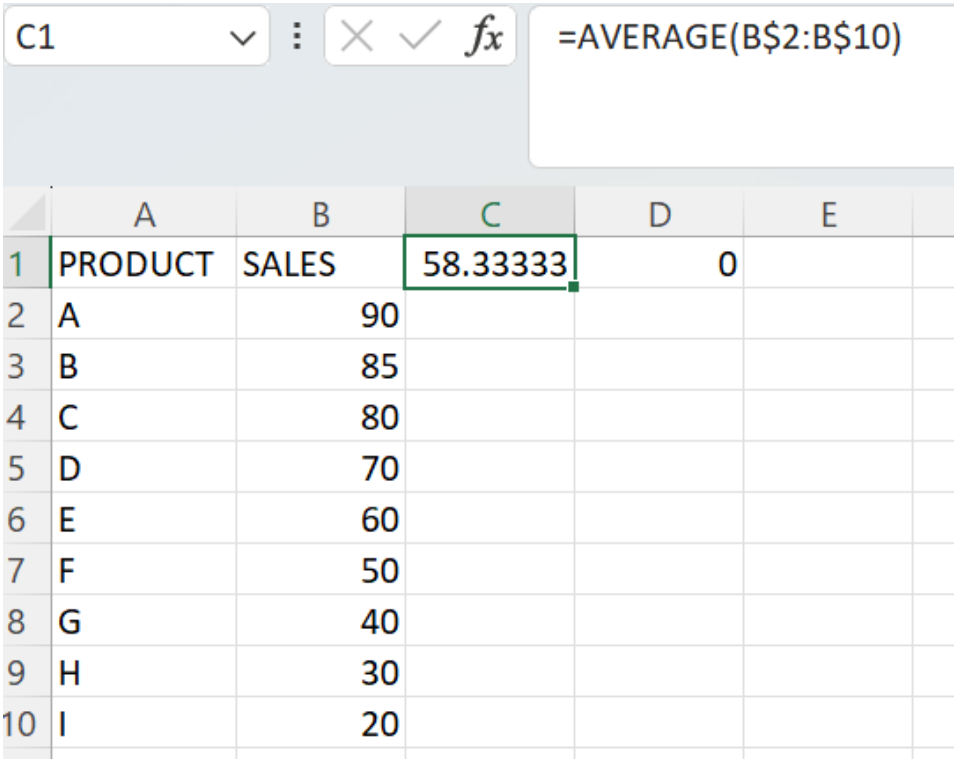
- *range* - Range to apply criteria to.
- *criteria* - Criteria to apply.
- *sum_range* - [optional] Range to sum. If omitted, cells in range are summed.



	A	B	C	D	E	F
1	PRODUCT	SALES	385	0		
2	A	90				
3	B	85				
4	C	80				
5	D	70				
6	E	60				
7	F	50				
8	G	40				
9	H	30				
10	I	20				

4. AVERAGE

The AVERAGE function calculates the average of numbers provided as arguments.



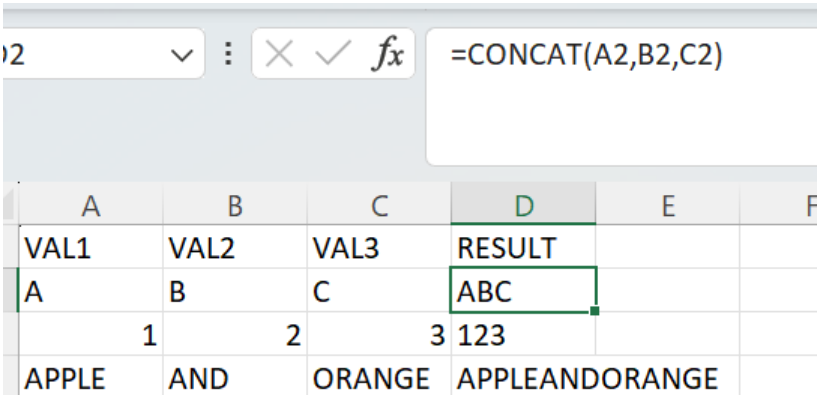
	A	B	C	D	E
1	PRODUCT	SALES	58.33333	0	
2	A	90			
3	B	85			
4	C	80			
5	D	70			
6	E	60			
7	F	50			
8	G	40			
9	H	30			
10	I	20			

5. CONCAT

The Excel CONCAT function concatenates (joins) values supplied as references.

Syntax

```
=CONCAT(text1,[text2],...)
```



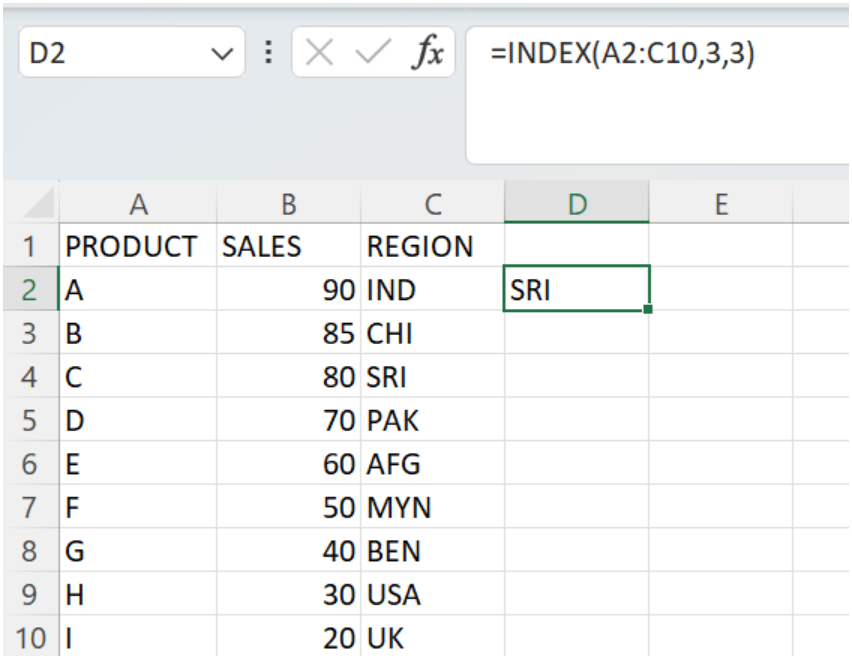
6. INDEX

The Excel INDEX function returns the value at a given location in a range or array.

Syntax

```
=INDEX(array,row_num,[col_num],[area_num])
```

- *array* - A range of cells, or an array constant.
- *row_num* - The row position in the reference or array.
- *col_num* - [optional] The column position in the reference or array.
- *area_num* - [optional] The range in reference that should be used.



	A	B	C	D	E
1	PRODUCT	SALES	REGION		
2	A	90	IND	SRI	
3	B	85	CHI		
4	C	80	SRI		
5	D	70	PAK		
6	E	60	AFG		
7	F	50	MYN		
8	G	40	BEN		
9	H	30	USA		
10	I	20	UK		

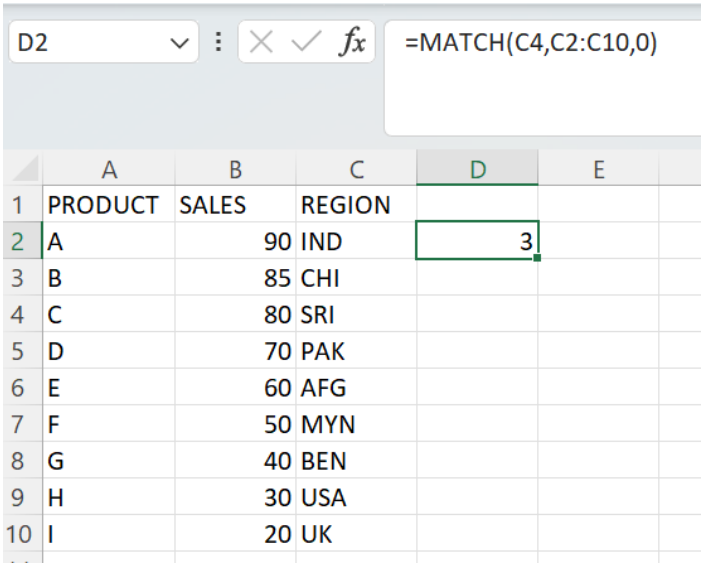
7.MATCH

MATCH is an Excel function used to locate the position of a lookup value in a row, column, or table.

Syntax

```
=MATCH(lookup_value,lookup_array,[match_type])
```

- *lookup_value* - The value to match in lookup_array.
- *lookup_array* - A range of cells or an array reference.
- *match_type* - [optional] 1 = exact or next smallest (default), 0 = exact match, -1 = exact or next largest.



	A	B	C	D	E
1	PRODUCT	SALES	REGION		
2	A	90	IND	3	
3	B	85	CHI		
4	C	80	SRI		
5	D	70	PAK		
6	E	60	AFG		
7	F	50	MYN		
8	G	40	BEN		
9	H	30	USA		
10	I	20	UK		

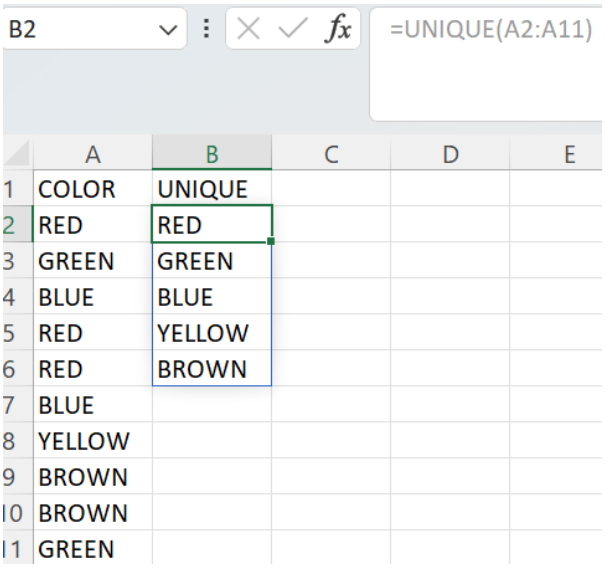
8. UNIQUE

The Excel UNIQUE function returns a list of unique values in a list or range.

Syntax

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

- *array* - Range or array from which to extract unique values.
- *by_col* - [optional] How to compare and extract. By row = FALSE (default); by column = TRUE.
- *exactly_once* - [optional] TRUE = values that occur once, FALSE= all unique values (default).


B2
=UNIQUE(A2:A11)
A
B
C
D
E
1 COLOR UNIQUE
2 RED RED
3 GREEN GREEN
4 BLUE BLUE
5 RED YELLOW
6 RED BROWN
7 BLUE
8 YELLOW
9 BROWN
10 BROWN
11 GREEN

9. IFS

The Excel IFS function can run multiple tests and return a value corresponding to the first TRUE result.

Syntax

```
=IFS(test1,value1,[test2, value2],...)
```

- *test1* - First logical test.
- *value1* - Result when test1 is TRUE.
- *test2, value2* - [optional] Second test/value pair.

C2 : =IFS(B2>=80,"A",B2>=60,"B",B2>=30,"C",B2<30,"D")

	A	B	C	D	E	F	G	H
1	NAME	SCORE	GRADE					
2	A	10	D					
3	B	90	A					
4	C	100	A					
5	D	70	B					
6	E	60	B					
7	F	40	C					
8	G	30	C					
9	H	20	D					
10	I	80	A					

10. COUNTIFS

The Excel COUNTIFS function returns the count of cells in a range that meet one or more conditions.

Syntax

```
=COUNTIFS(range1,criteria1,[range2],[criteria2],...)
```

- *range1* - The first range to evaluate.
- *criteria1* - The criteria to use on range1.
- *range2* - [optional] The second range to evaluate.
- *criteria2* - [optional] The criteria to use on range2.

E1 =COUNTIFS(B2:B10,">80",C2:C10,">70",D2:D10,">90")

	A	B	C	D	E	F	G	H
1	STUDENT	PHYSICS	CHEMISTRY	MATHS	1			
2	A	100	80	90				
3	B	75	60	70				
4	C	50	80	70				
5	D	25	50	70				
6	E	40	50	60				
7	F	60	60	70				
8	G	90	80	100				
9	H	60	65	70				
10	I	80	70	100				

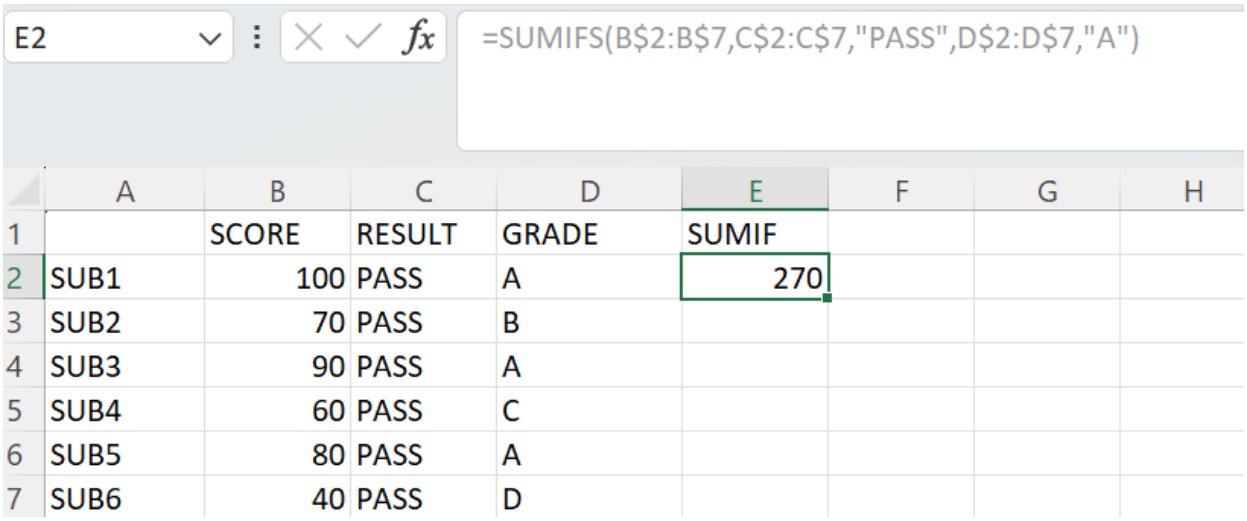
11. SUMIFS

The Excel SUMIFS function returns the sum of cells that meet multiple conditions, referred to as criteria.

Syntax

```
=SUMIFS(sum_range,range1,criteria1,[range2],[criteria2],...)
```

- *sum_range* - The range to be summed.
- *range1* - The first range to evaluate.
- *criteria1* - The criteria to use on range1.
- *range2* - [optional] The second range to evaluate.
- *criteria2* - [optional] The criteria to use on range2.



	A	B	C	D	E	F	G	H
1		SCORE	RESULT	GRADE	SUMIF			
2	SUB1	100	PASS	A	270			
3	SUB2	70	PASS	B				
4	SUB3	90	PASS	A				
5	SUB4	60	PASS	C				
6	SUB5	80	PASS	A				
7	SUB6	40	PASS	D				

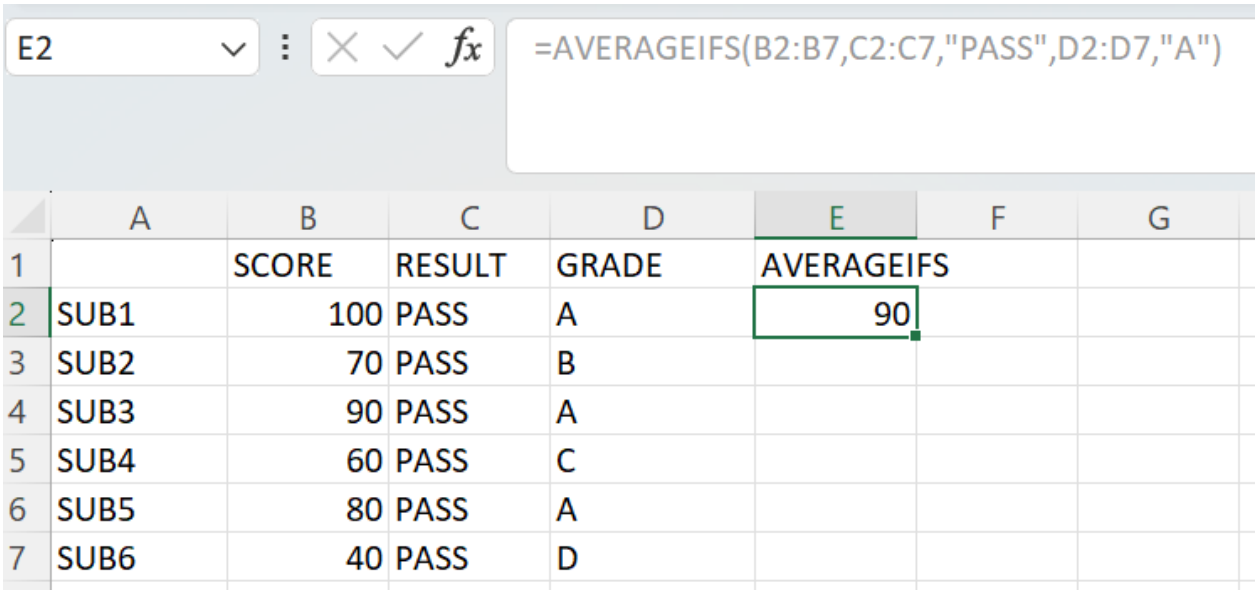
12. AVERAGEIFS

The Excel AVERAGEIFS function returns the average of cells that meet multiple conditions, referred to as criteria.

Syntax

```
=AVERAGEIFS(avg_rng,range1,criteria1,[range2],[criteria2],...)
```

- *avg_rng* - The range to average.
- *range1* - The first range to evaluate.
- *criteria1* - The criteria to use on range1.
- *range2* - [optional] The second range to evaluate.
- *criteria2* - [optional] The criteria to use on range2.



	A	B	C	D	E	F	G
1		SCORE	RESULT	GRADE	AVERAGEIFS		
2	SUB1	100	PASS	A	90		
3	SUB2	70	PASS	B			
4	SUB3	90	PASS	A			
5	SUB4	60	PASS	C			
6	SUB5	80	PASS	A			
7	SUB6	40	PASS	D			

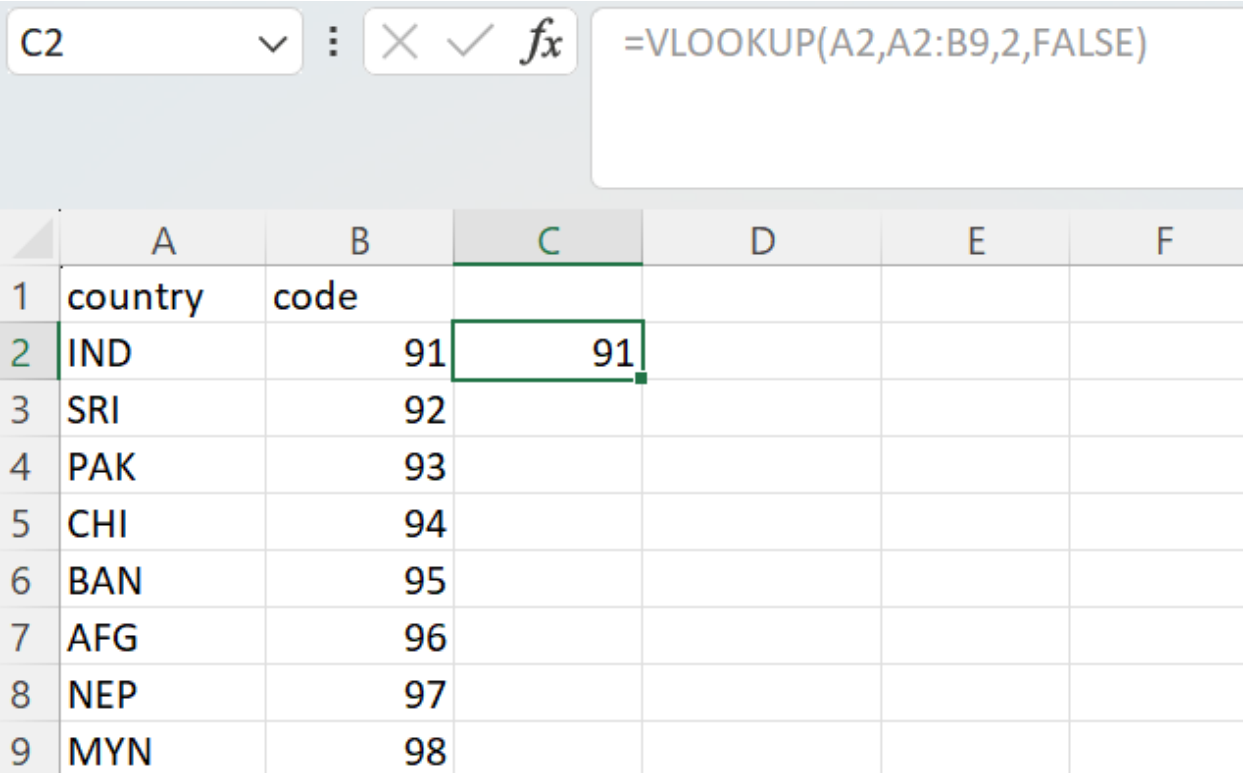
13. VLOOKUP

The Excel VLOOKUP function is used to retrieve information from a table using a lookup value. The lookup values must appear in the *first* column of the table, and the information to retrieve is specified by *column number*.

Syntax

```
=VLOOKUP(lookup_value,table_array,column_index_num,[range_lookup])
```

- *lookup_value* - The value to look for in the first column of a table.
- *table_array* - The table from which to retrieve a value.
- *column_index_num* - The column in the table from which to retrieve a value.
- *range_lookup* - [optional] TRUE = approximate match (default). FALSE = exact match.



	A	B	C	D	E	F
1	country	code				
2	IND	91	91			
3	SRI	92				
4	PAK	93				
5	CHI	94				
6	BAN	95				
7	AFG	96				
8	NEP	97				
9	MYN	98				

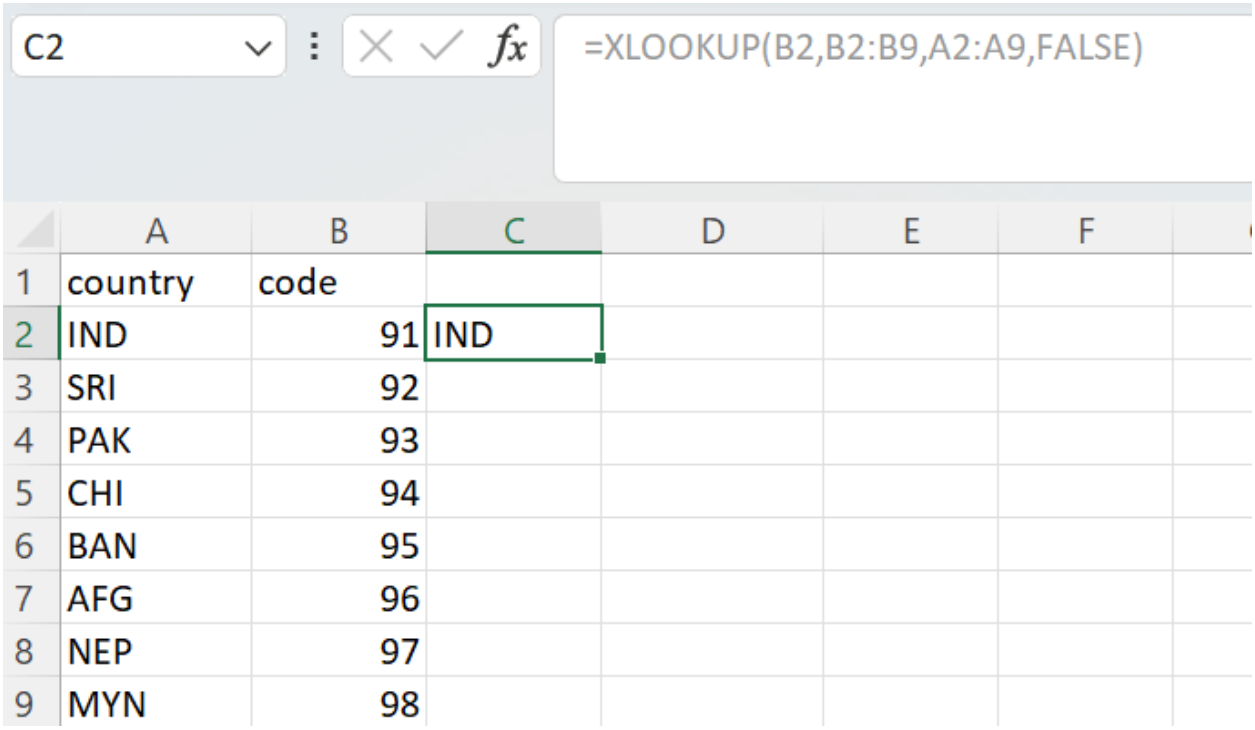
14. XLOOKUP

The Excel XLOOKUP function is a modern and flexible replacement for older functions like VLOOKUP, HLOOKUP, and LOOKUP. XLOOKUP supports approximate and exact matching, wildcards (* ?) for partial matches, and lookups

in vertical or horizontal ranges.

```
=XLOOKUP(lookup,lookup_array,return_array,[not_found],[match_mode],
[search_mode])
```

- *lookup* - The lookup value.
- *lookup_array* - The array or range to search.
- *return_array* - The array or range to return.
- *not_found* - [optional] Value to return if no match found.
- *match_mode* - [optional] 0 = exact match (default), -1 = exact match or next smallest, 1 = exact match or next larger, 2 = wildcard match.
- *search_mode* - [optional] 1 = search from first (default), -1 = search from last, 2 = binary search ascending, -2 = binary search descending.



15. HLOOKUP

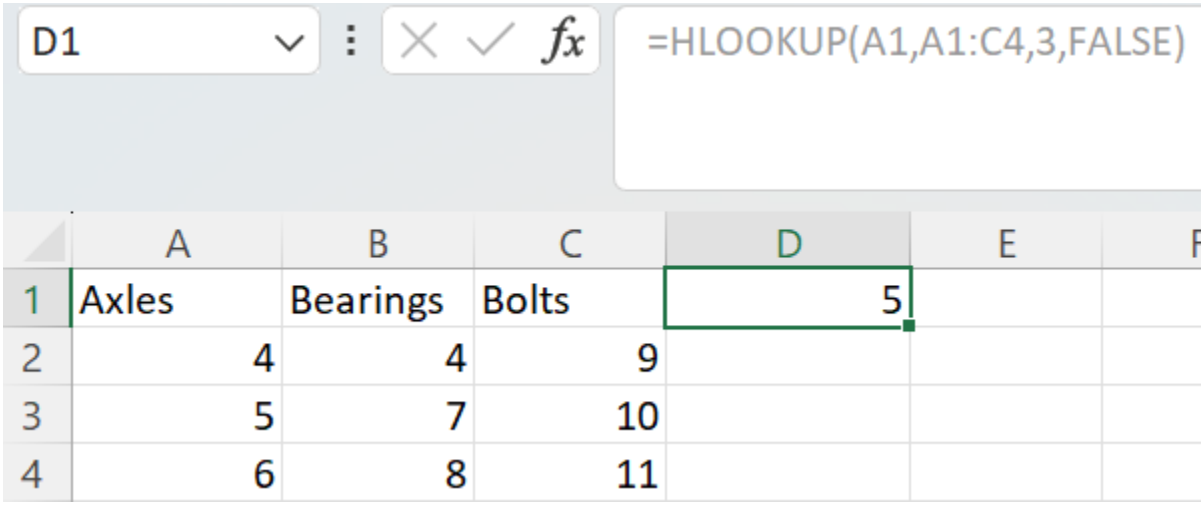
The Excel HLOOKUP function finds and retrieve a value from data in a horizontal

table. The "H" in HLOOKUP stands for "horizontal", and lookup values must appear in the first row of the table, moving horizontally to the right.

Syntax

```
=HLOOKUP(lookup_value,table_array,row_index,[range_lookup])
```

- *lookup_value* - The value to look up.
- *table_array* - The table from which to retrieve data.
- *row_index* - The row number from which to retrieve data.
- *range_lookup* - [optional] A Boolean to indicate exact match or approximate match. Default = TRUE = approximate match.



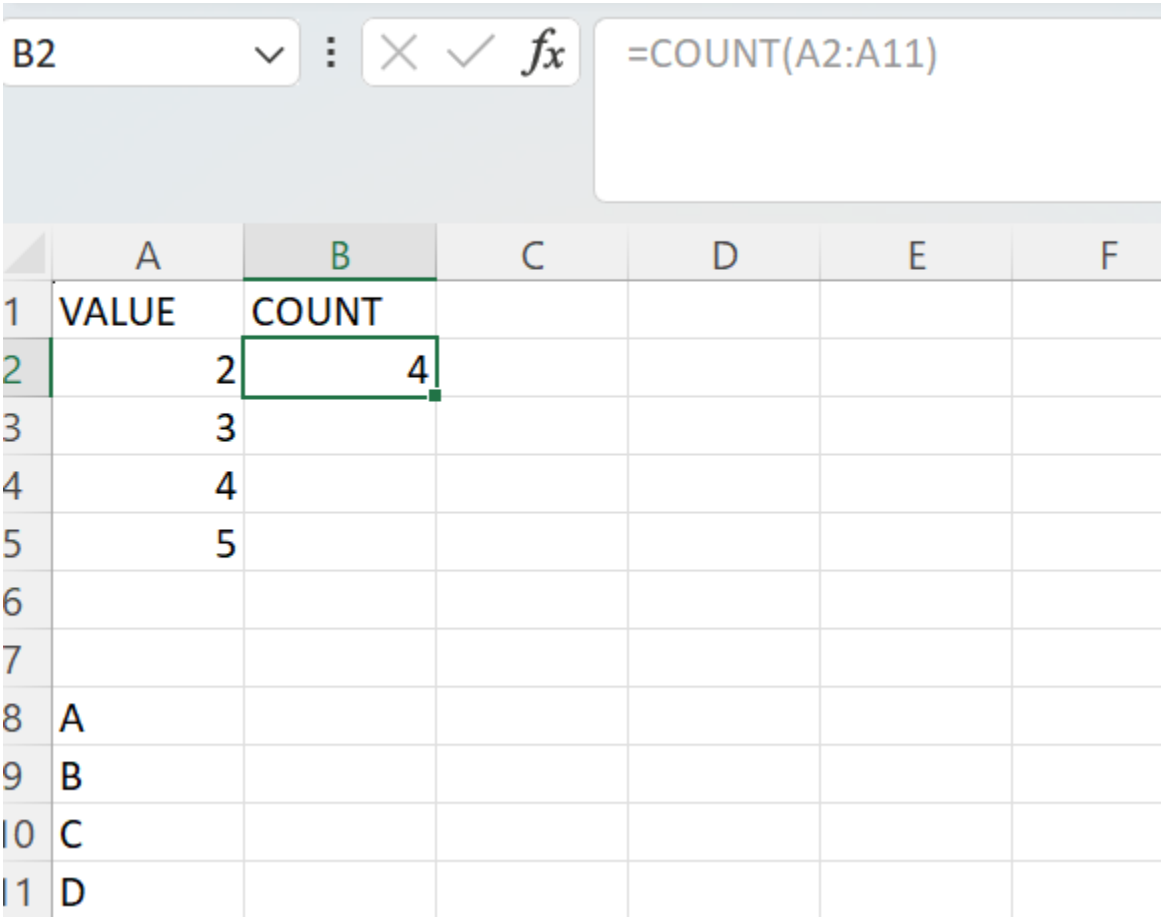
16. COUNT

The Excel COUNT function returns a count of values that are numbers. Empty cells and text values are ignored.

Syntax

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

- *value1* - An item, cell reference, or range.
- *value2* - [optional] An item, cell reference, or range.



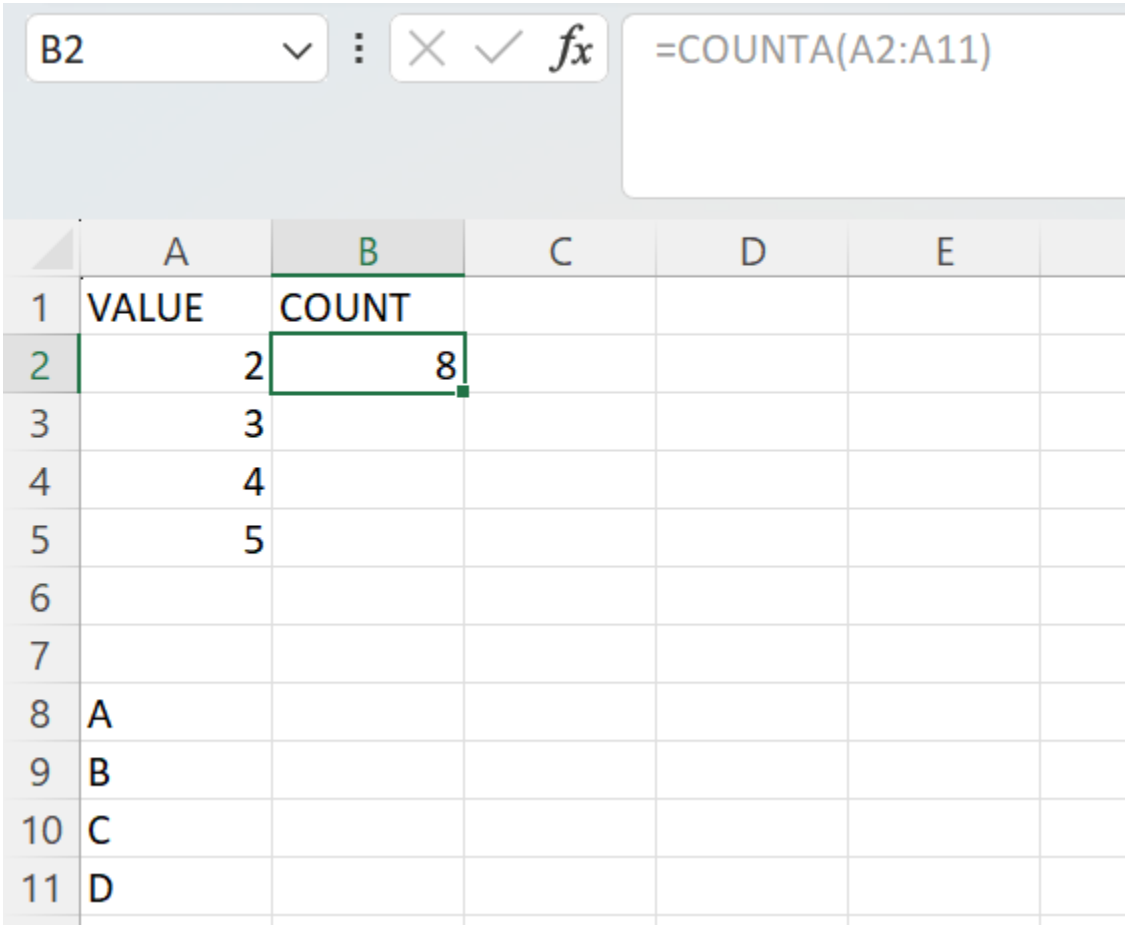
17. COUNTA

The Excel COUNTA function returns the count of cells that contain numbers, text, logical values, error values, and empty text (""). COUNTA does not count empty cells.

Syntax

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

- *value1* - An item, cell reference, or range.
- *value2* - [optional] An item, cell reference, or range.



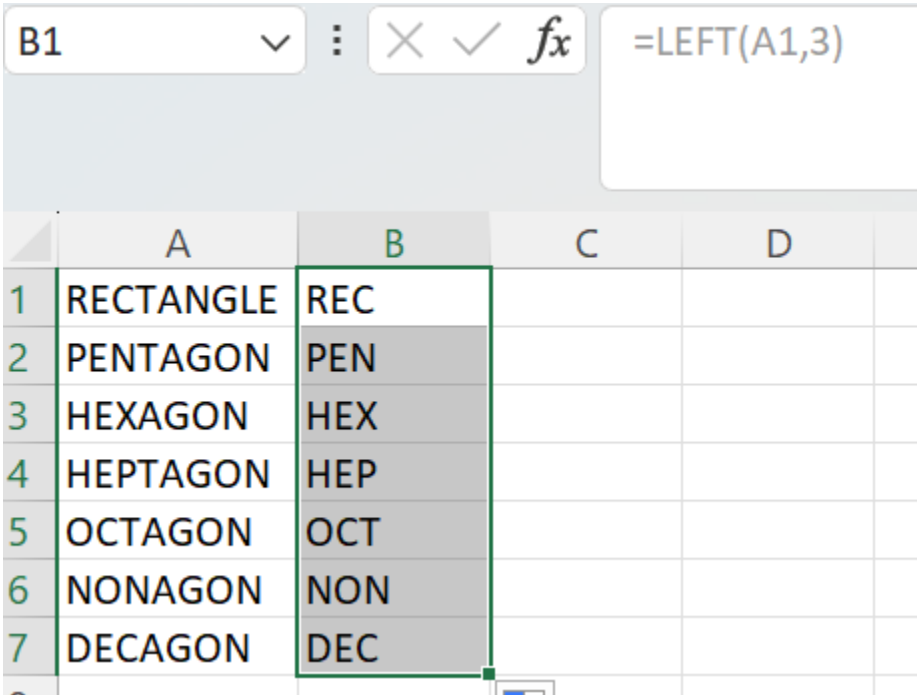
18. LEFT

The Excel LEFT function extracts a given number of characters from the left side of a supplied text string. For example, =LEFT("apple",3) returns "app".

Syntax

```
=LEFT(text,[num_chars])
```

- *text* - The text from which to extract characters.
- *num_chars* - [optional] The number of characters to extract, starting on the left. Default = 1.



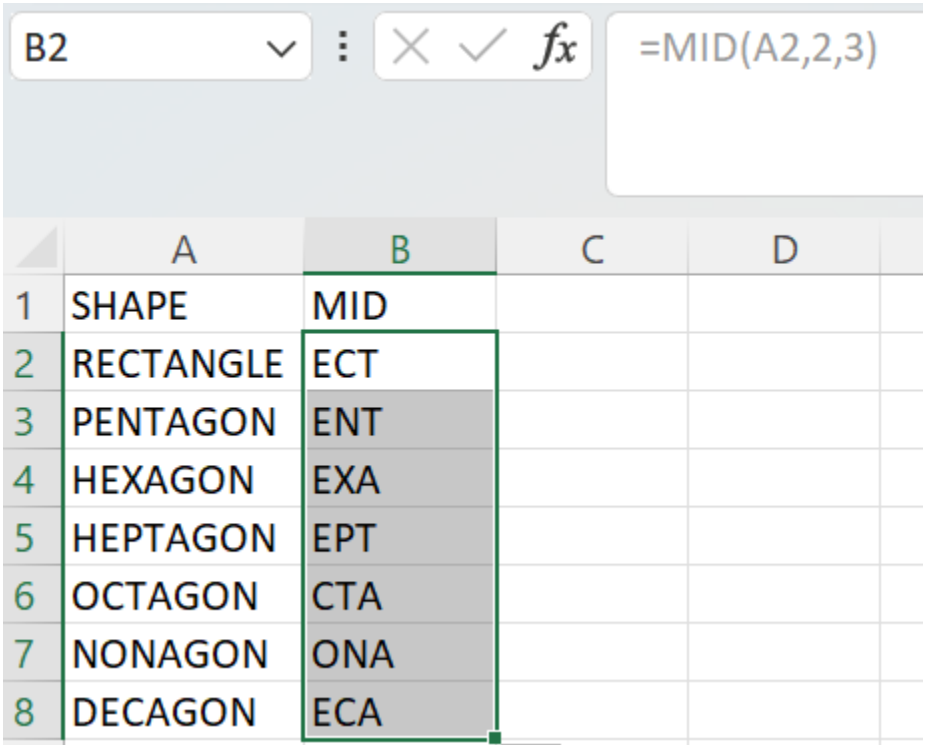
19. MID

The Excel MID function extracts a given number of characters from the middle of a supplied text string based on the provided starting location. For example, =MID("apple",2,3) returns "ppl".

Syntax

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

- *text* - The text to extract from.
- *start_num* - The location of the first character to extract.
- *num_chars* - The number of characters to extract.

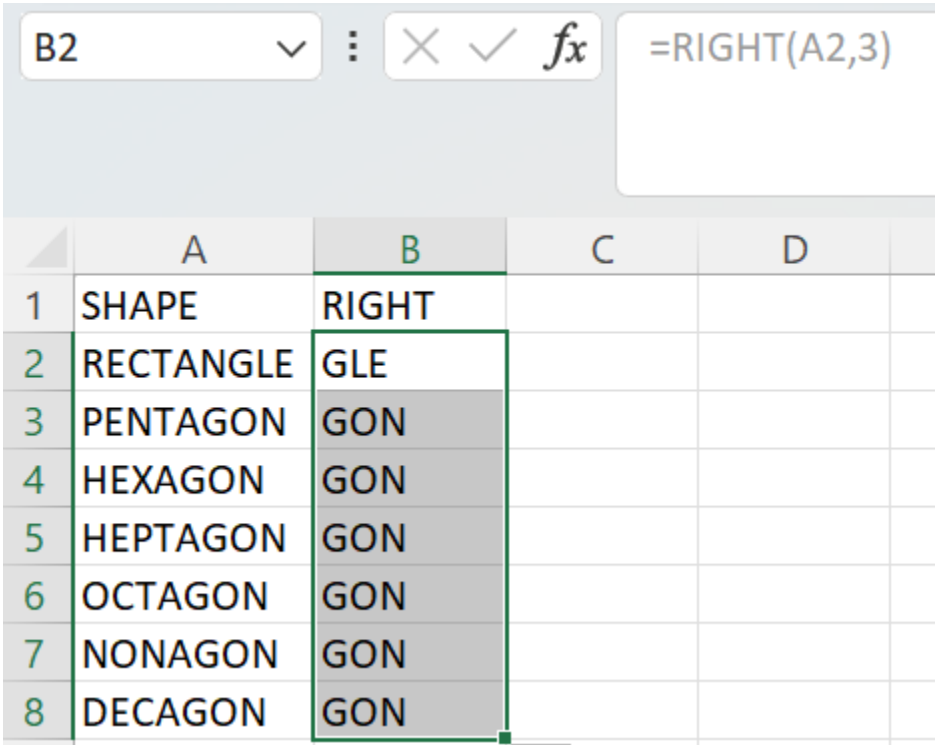


20. RIGHT

The Excel RIGHT function extracts a given number of characters from the *right* side of a supplied text string. For example, =RIGHT("apple",3) returns "ple".

```
=RIGHT(text,[num_chars])
```

- *text* - The text from which to extract characters on the right.
- *num_chars* - [optional] The number of characters to extract, starting on the right. Default = 1.



	A	B	C	D
1	SHAPE	RIGHT		
2	RECTANGLE	GLE		
3	PENTAGON	GON		
4	HEXAGON	GON		
5	HEPTAGON	GON		
6	OCTAGON	GON		
7	NONAGON	GON		
8	DECAGON	GON		

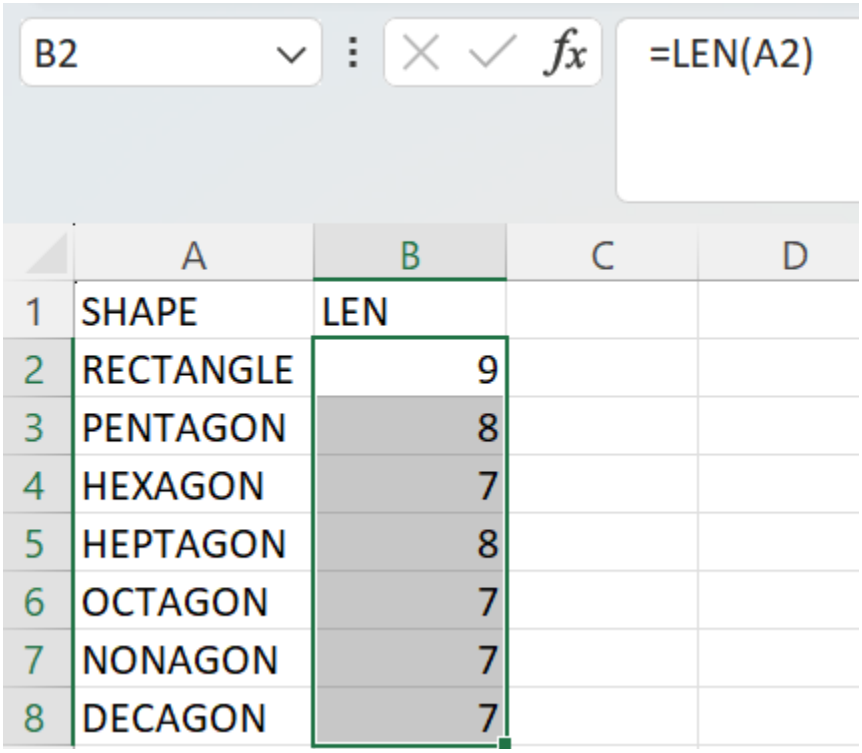
21. LEN

The Excel LEN function returns the length of a given text string as the number of characters. LEN will also count characters in numbers.

Syntax

```
=LEN(text)
```

- *text* - The text for which to calculate length.



	A	B	C	D
1	SHAPE	LEN		
2	RECTANGLE	9		
3	PENTAGON	8		
4	HEXAGON	7		
5	HEPTAGON	8		
6	OCTAGON	7		
7	NONAGON	7		
8	DECAGON	7		

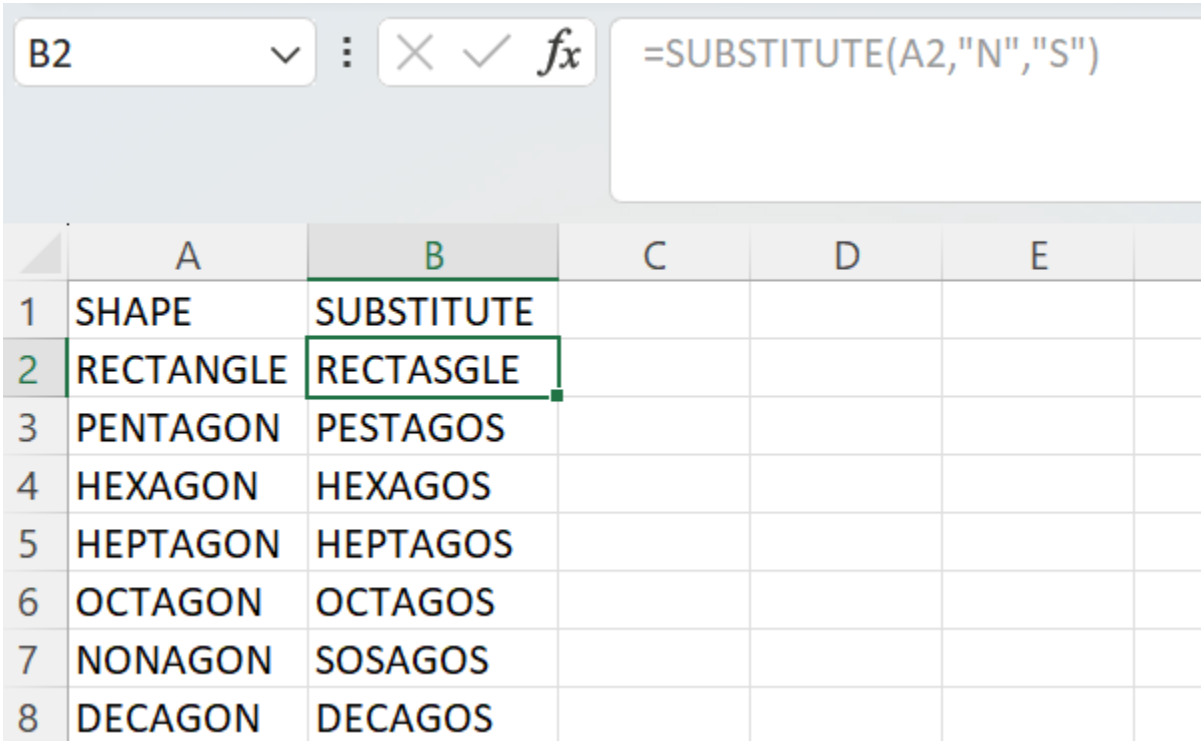
22. SUBSTITUTE

The Excel SUBSTITUTE function replaces text in a given string by matching.

Syntax

```
=SUBSTITUTE(text,old_text,new_text,[instance])
```

- *text* - The text to change.
- *old_text* - The text to replace.
- *new_text* - The text to replace with.
- *instance* - [optional] The instance to replace. If not supplied, all instances are replaced.



	A	B	C	D	E
1	SHAPE	SUBSTITUTE			
2	RECTANGLE	RECTASGLE			
3	PENTAGON	PESTAGOS			
4	HEXAGON	HEXAGOS			
5	HEPTAGON	HEPTAGOS			
6	OCTAGON	OCTAGOS			
7	NONAGON	SOSAGOS			
8	DECAGON	DECAGOS			

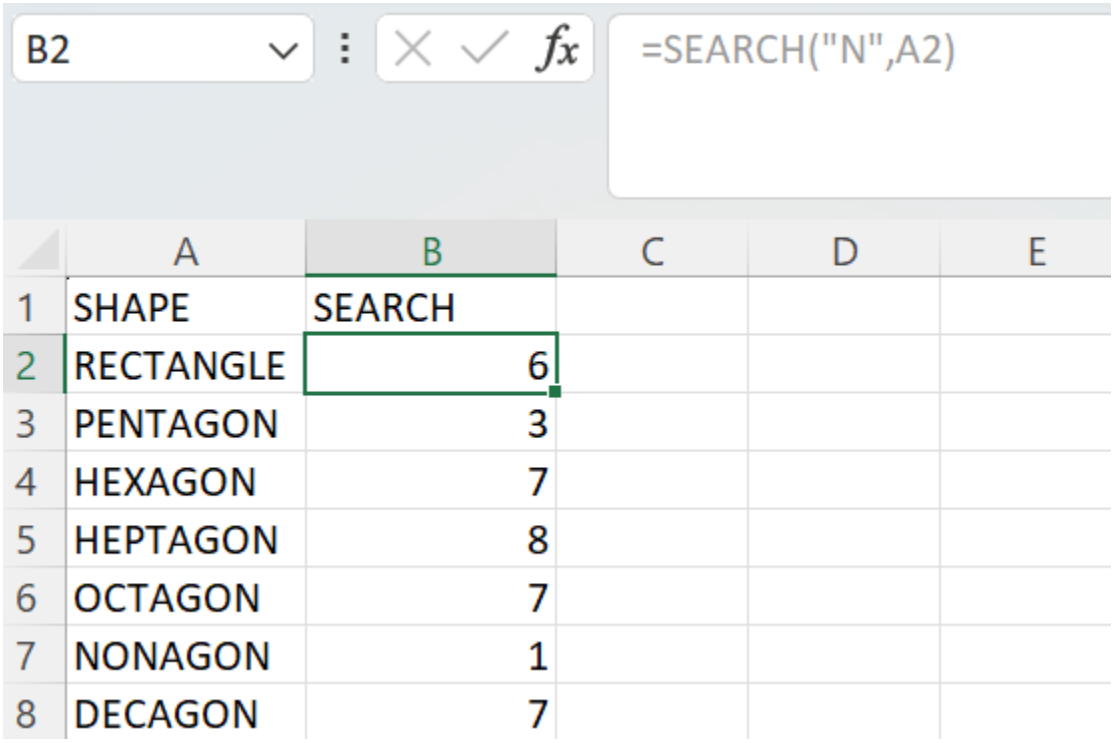
23. SEARCH

The Excel SEARCH function returns the location of one text string inside another.

Syntax

```
=SEARCH(find_text,within_text,[start_num])
```

- *find_text* - The substring to find.
- *within_text* - The text to search within.
- *start_num* - [optional] Starting position. Optional, defaults to 1.



	A	B	C	D	E
1	SHAPE	SEARCH			
2	RECTANGLE	6			
3	PENTAGON	3			
4	HEXAGON	7			
5	HEPTAGON	8			
6	OCTAGON	7			
7	NONAGON	1			
8	DECAGON	7			

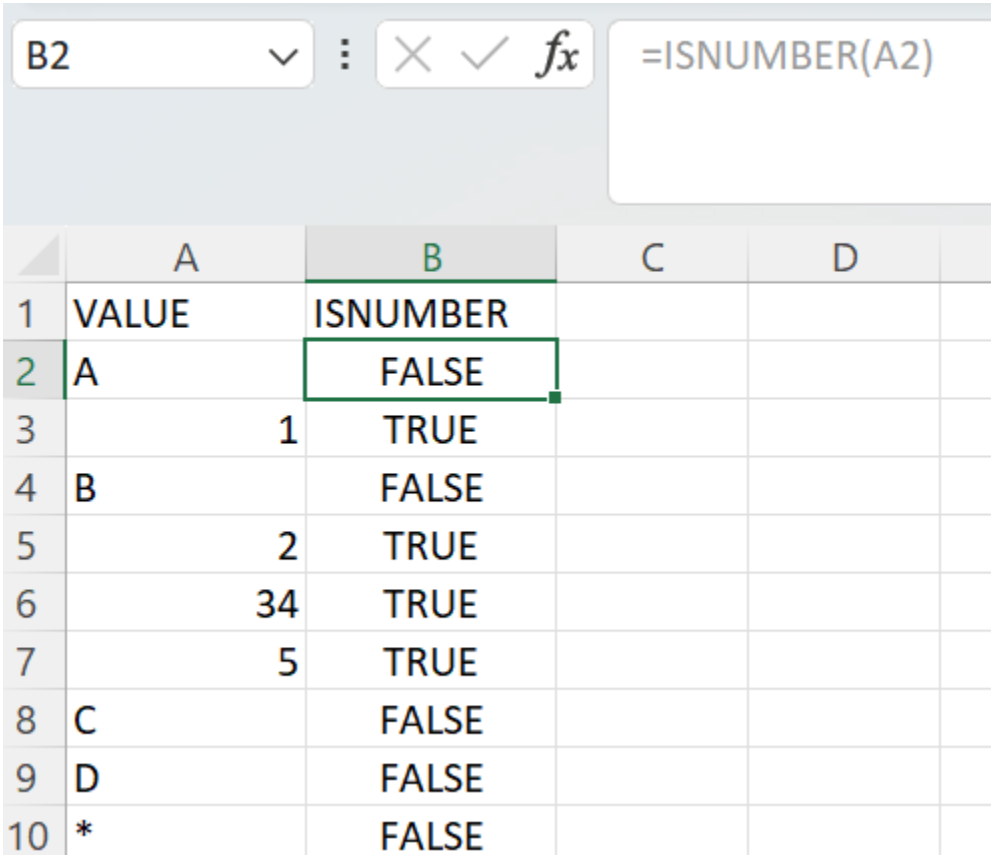
24. ISNUMBER

The Excel ISNUMBER function returns TRUE when a cell contains a number, and FALSE if not.

Syntax

```
=ISNUMBER(value)
```

- *value* - The value to check.



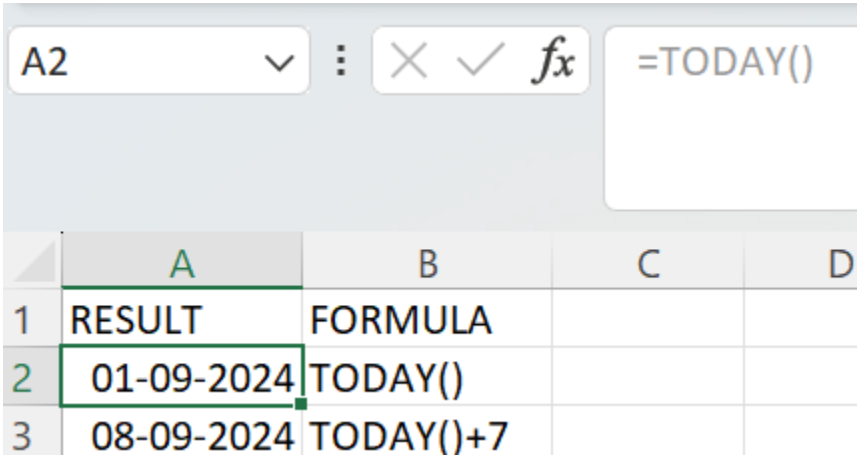
	A	B	C	D
1	VALUE	ISNUMBER		
2	A	FALSE		
3	1	TRUE		
4	B	FALSE		
5	2	TRUE		
6	34	TRUE		
7	5	TRUE		
8	C	FALSE		
9	D	FALSE		
10	*	FALSE		

25. TODAY

The Excel TODAY function returns the current date.

Syntax

```
=TODAY()
```

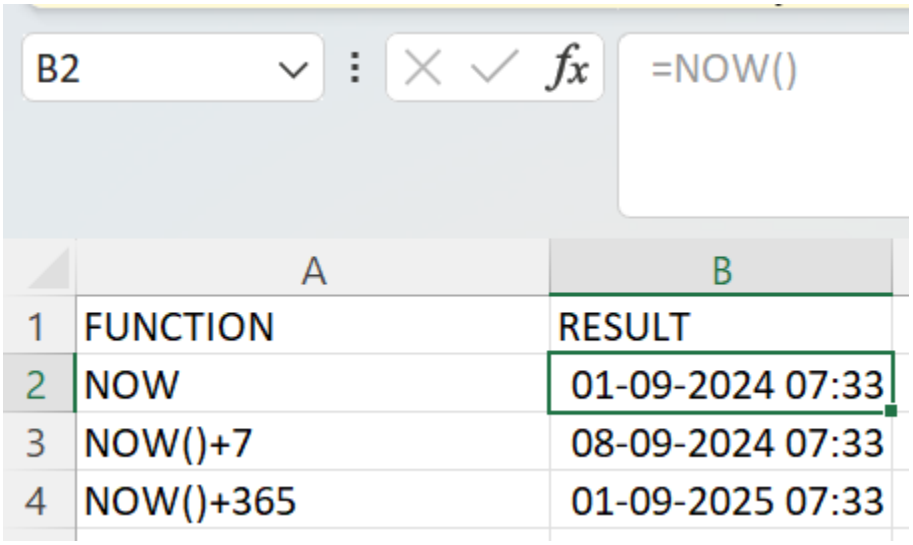


26. NOW

The Excel NOW function returns the current date and time.

Syntax

```
=NOW()
```



	A	B
1	FUNCTION	RESULT
2	NOW	01-09-2024 07:33
3	NOW()+7	08-09-2024 07:33
4	NOW()+365	01-09-2025 07:33

27. YEAR

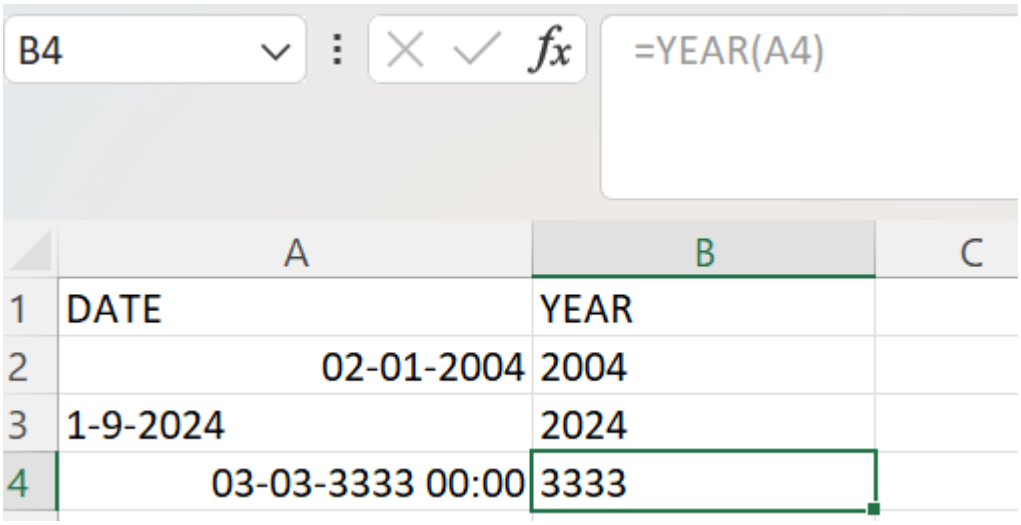
The Excel YEAR function returns the year component of a date as a 4-digit number.

The default start year in excel is 1900.In case date is displayed wrongly format cell into text after executing the function.

Syntax

```
=YEAR(date)
```

- *date* - A valid Excel date.



28. MONTH

The Excel MONTH function extracts the month from a given date as a number between 1 and 12.

Syntax

```
=MONTH(serial_number)
```

- *serial_number* - A valid Excel date.

date	month
01-01-2000	1
02-02-2222	2
03-03-3333	3

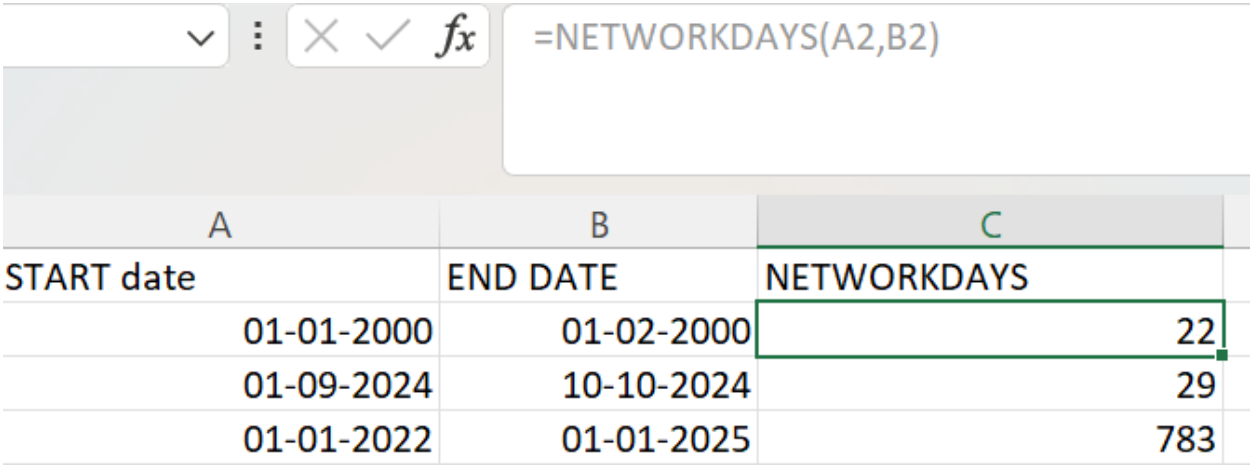
29. NETWORKDAYS

The Excel NETWORKDAYS function calculates the number of working days between two dates. NETWORKDAYS automatically excludes weekends (Saturday and Sunday) and can optionally exclude a list of holidays supplied as dates.

Syntax

```
=NETWORKDAYS(start_date,end_date,[holidays])
```

- *start_date* - The start date.
- *end_date* - The end date.
- *holidays* - [optional] A list of non-work days as dates.



	A	B	C
	START date	END DATE	NETWORKDAYS
	01-01-2000	01-02-2000	22
	01-09-2024	10-10-2024	29
	01-01-2022	01-01-2025	783

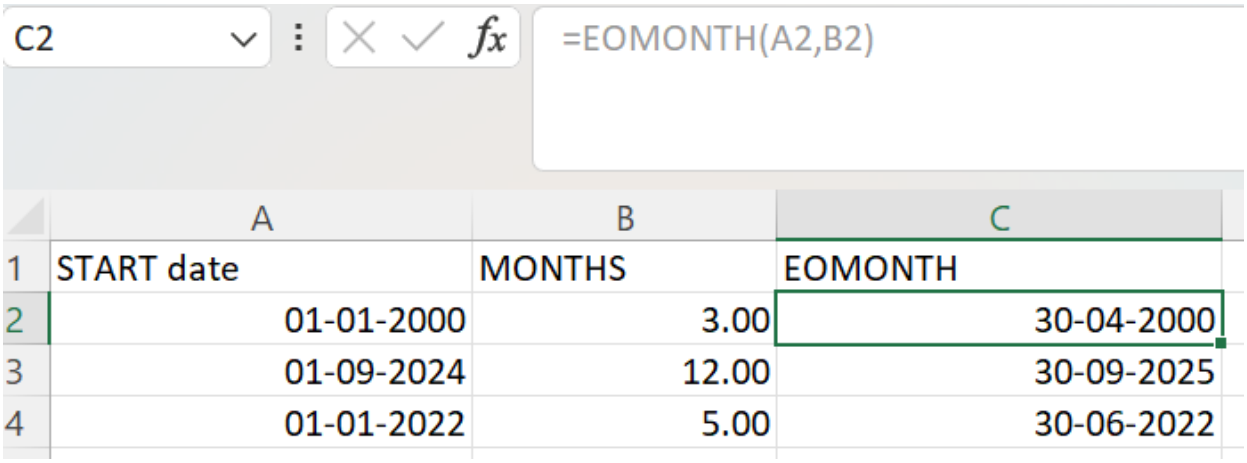
30. EOMONTH

The Excel EOMONTH function returns the last day of the month, n months in the past or future. You can use EOMONTH to calculate expiration dates, due dates, and other dates that need to land on the last day of a month. Use a positive value for months to move forward in time, and a negative number to move back in time.

Syntax

```
=EOMONTH(start_date,months)
```

- *start_date* - A date that represents the start date in a valid Excel serial number format.
- *months* - The number of months before or after start_date.



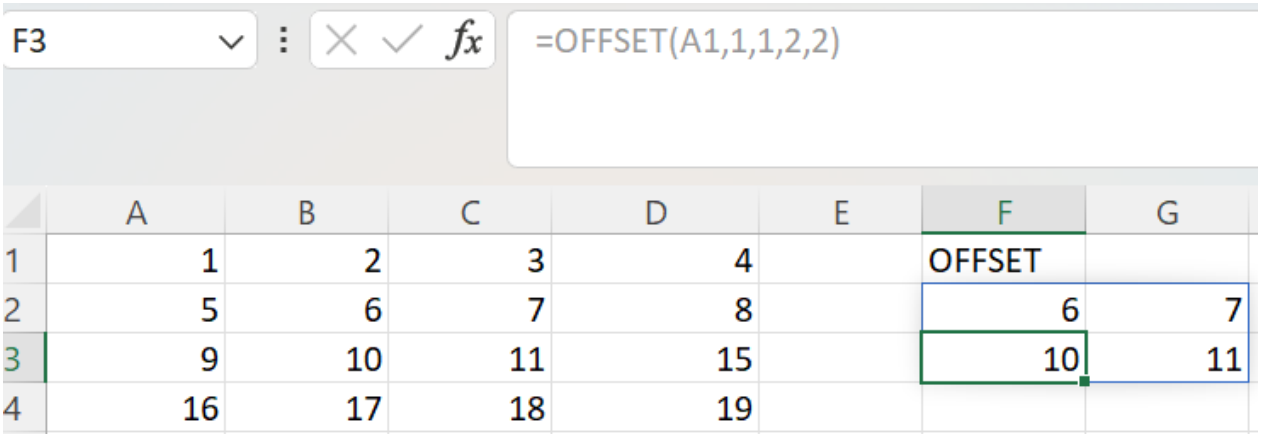
	A	B	C
1	START date	MONTHS	EOMONTH
2	01-01-2000	3.00	30-04-2000
3	01-09-2024	12.00	30-09-2025
4	01-01-2022	5.00	30-06-2022

31. OFFSET

The Excel OFFSET function returns a reference to a range constructed with five inputs: (1) a starting point, (2) a row offset, (3) a column offset, (4) a height in rows, (5) a width in columns.

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

- *reference* - The starting point, supplied as a cell reference or range.
- *rows* - The number of rows to offset below the starting reference.
- *cols* - The number of columns to offset to the right of the starting reference.
- *height* - [optional] The height in rows of the returned reference.
- *width* - [optional] The width in columns of the returned reference.



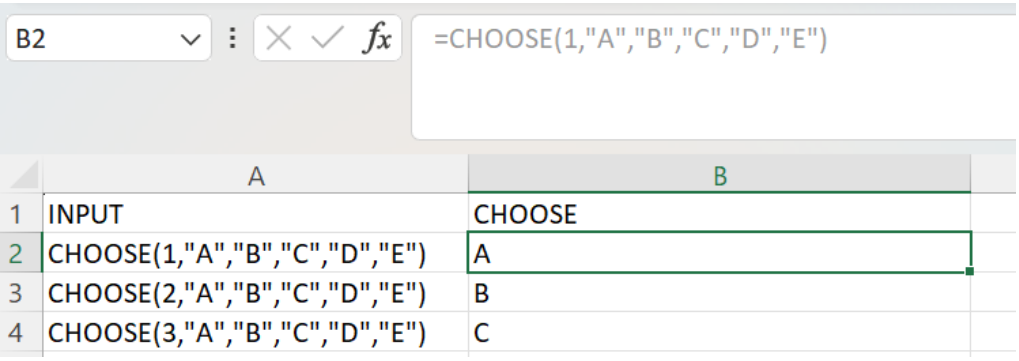
32. CHOOSE

The Excel CHOOSE function returns a value from a list using a given position or index. For example, =CHOOSE(2,"red","blue","green") returns "blue", since blue is the 2nd value listed after the index number.

Syntax

```
=CHOOSE(index_num,value1,[value2],...)
```

- *index_num* - The value to choose. A number between 1 and 254.
- *value1* - The first value from which to choose.
- *value2* - [optional] The second value from which to choose.



	A	B
1	INPUT	CHOOSE
2	CHOOSE(1,"A","B","C","D","E")	A
3	CHOOSE(2,"A","B","C","D","E")	B
4	CHOOSE(3,"A","B","C","D","E")	C

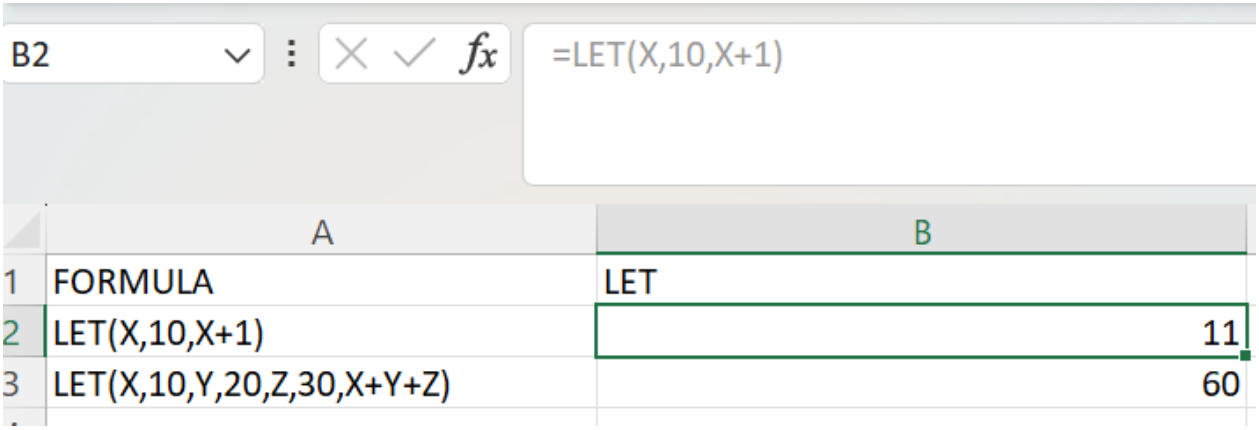
33. LET

The Excel LET function lets you define named variables in a formula.

Syntax

```
=LET(name1,value1,[name2/value2],...,result)
```

- *name1* - First name to assign. Must begin with a letter.
- *value1* - The value or calculation to assign to name 1.
- *name2/value2* - [optional] Second name and value. Entered as a pair of arguments.
- *result* - A calculation or a variable previously calculated.

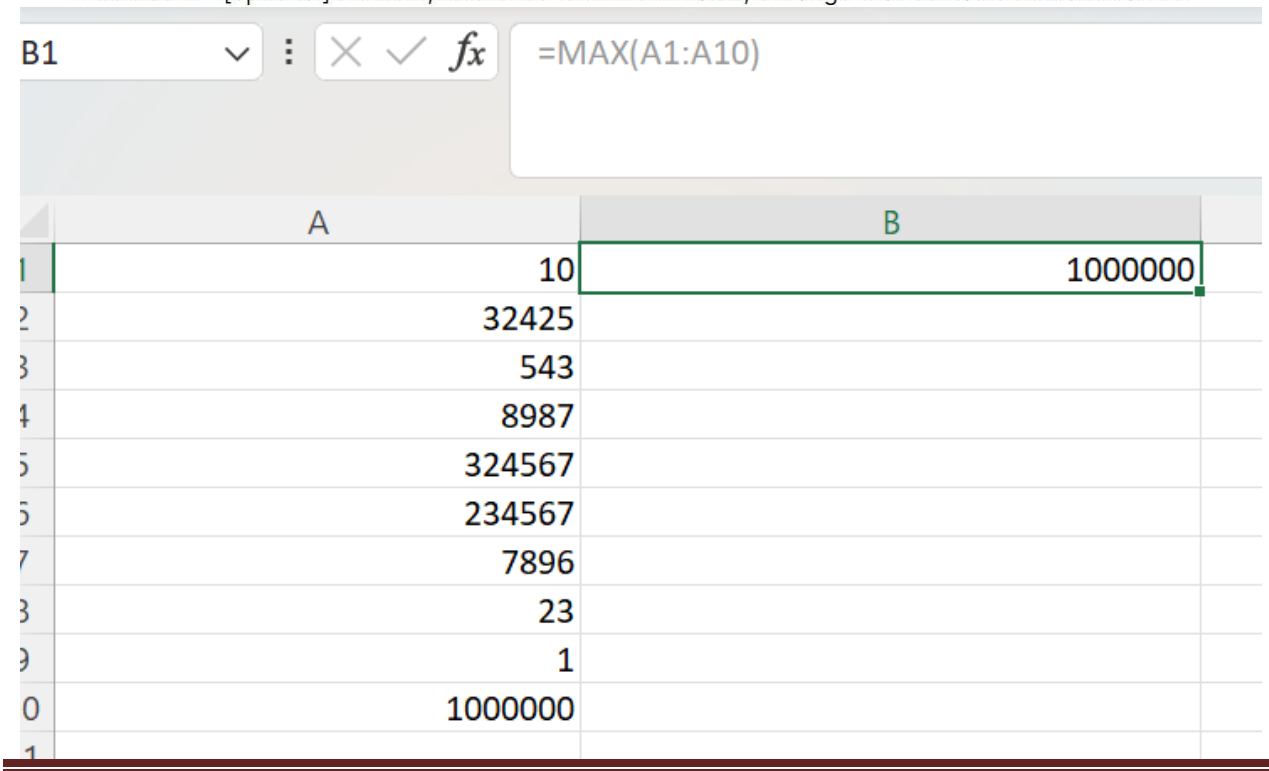


34. MAX

The Excel MAX function returns the largest numeric value in the data provided.

```
=MAX(number1,[number2],...)
```

- *number1* - Number, reference to numeric value, or range that contains numeric values.
- *number2* - [optional] Number, reference to numeric value, or range that contains numeric values.

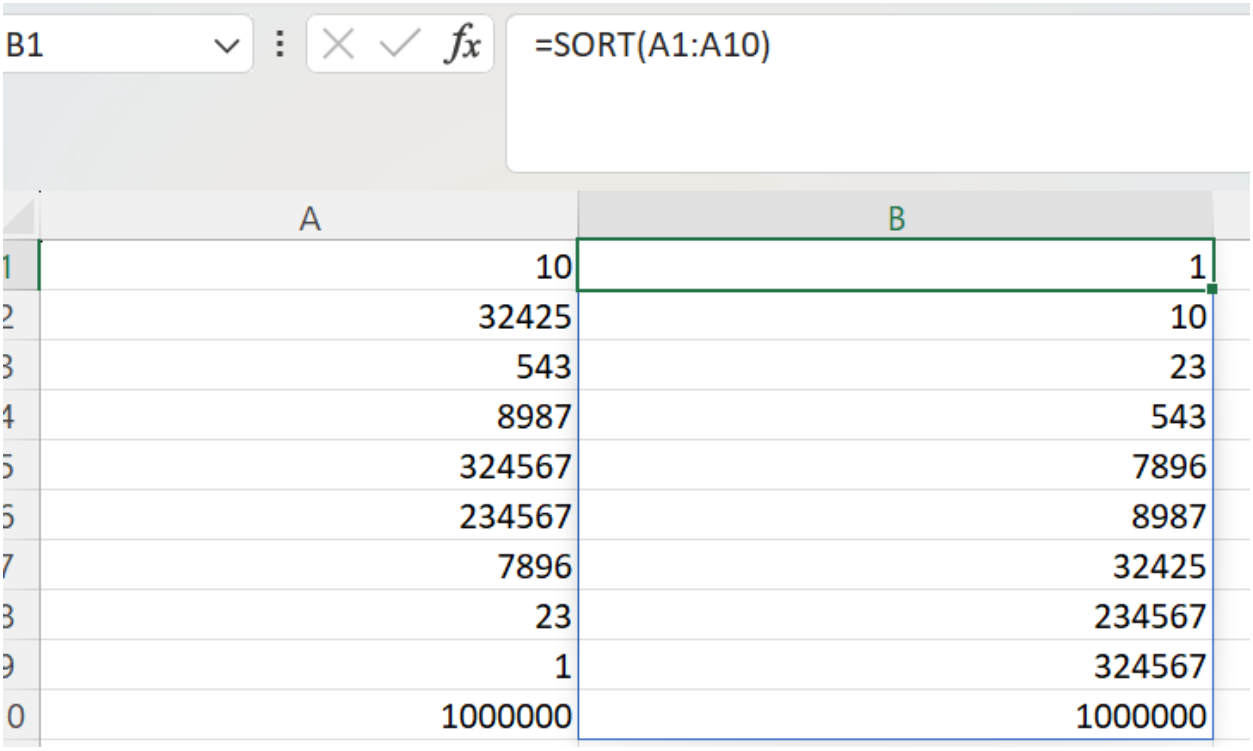


35. SORT

The Excel SORT function sorts the contents of a range or array in ascending or descending order.

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

- *array* - Range or array to sort.
- *sort_index* - [optional] Column index to use for sorting. Default is 1.
- *sort_order* - [optional] 1 = Ascending, -1 = Descending. Default is ascending order.
- *by_col* - [optional] TRUE = sort by column. FALSE = sort by row. Default is FALSE.



	A	B
1	10	1
2	32425	10
3	543	23
4	8987	543
5	324567	7896
6	234567	8987
7	7896	32425
8	23	234567
9	1	324567
10	1000000	1000000

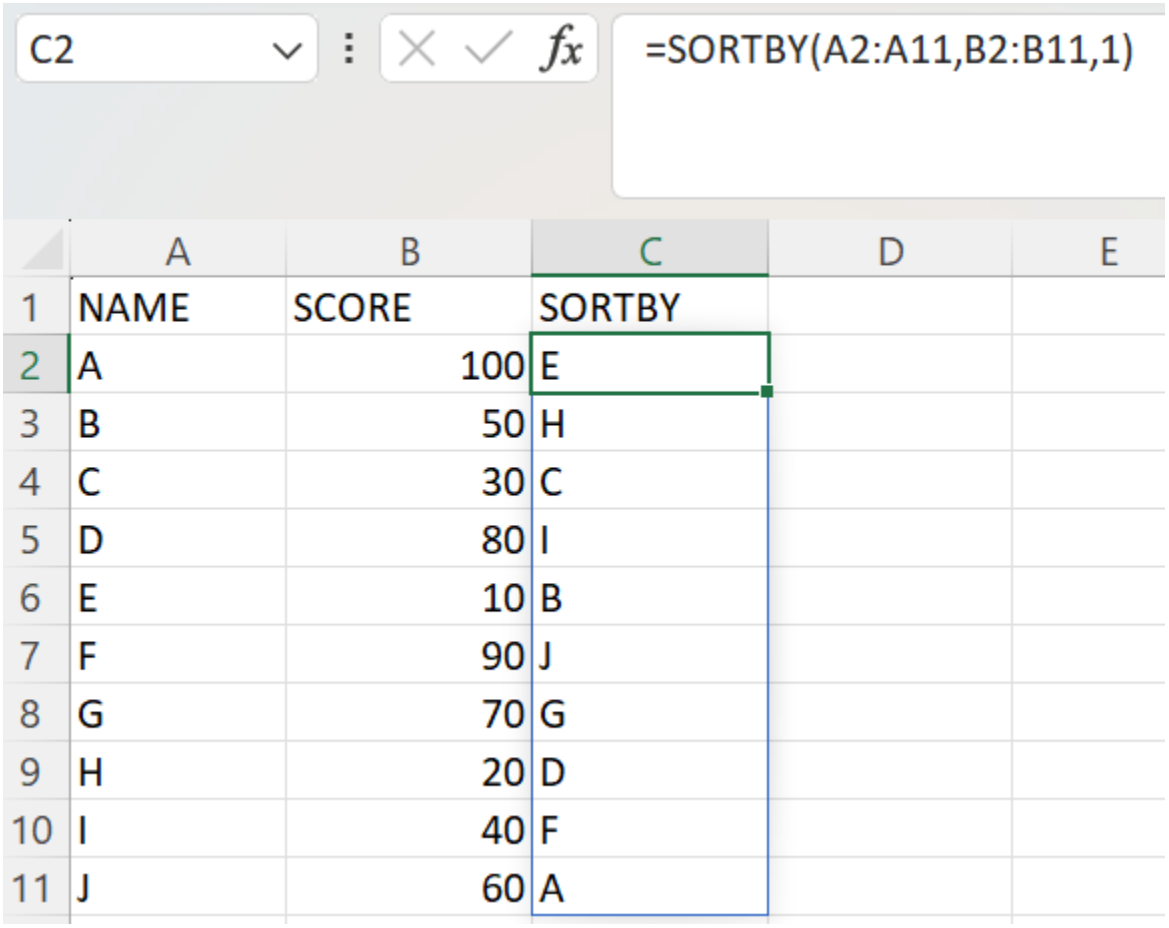
36. SORTBY

The Excel SORTBY function sorts the contents of a range or array based on the values from another range or array.

Syntax

```
=SORTBY(array,by_array,[sort_order],[array/order],...)
```

- *array* - Range or array to sort.
- *by_array* - Range or array to sort by.
- *sort_order* - [optional] Sort order. 1 = ascending (default), -1 = descending.
- *array/order* - [optional] Additional array and sort order pairs.



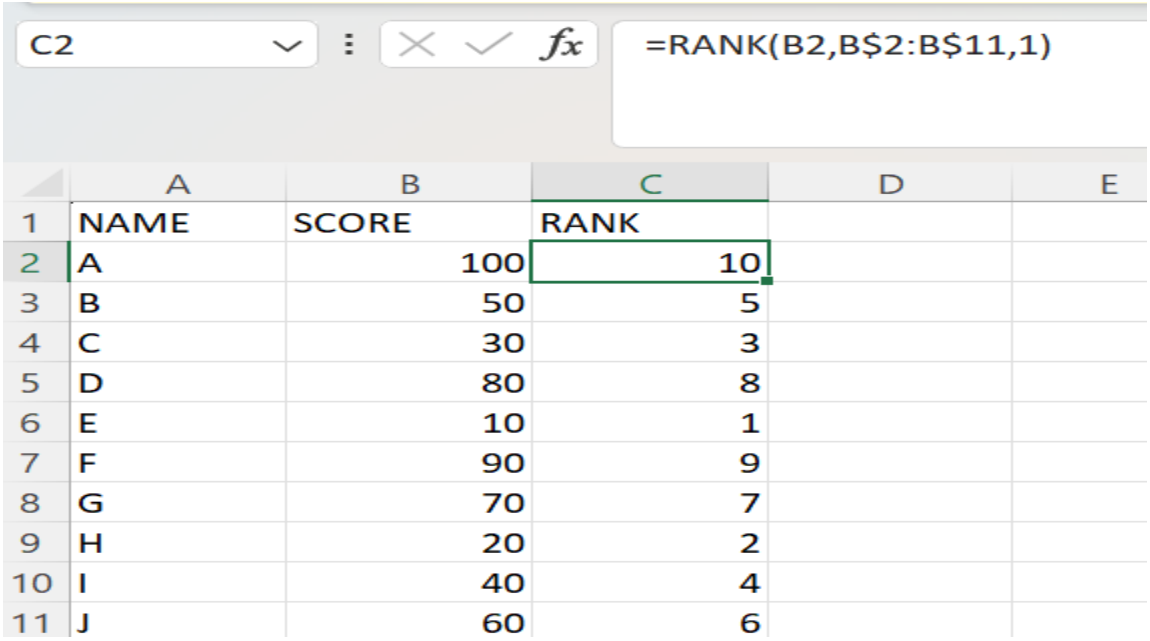
	A	B	C	D	E
1	NAME	SCORE	SORTBY		
2	A	100	E		
3	B	50	H		
4	C	30	C		
5	D	80	I		
6	E	10	B		
7	F	90	J		
8	G	70	G		
9	H	20	D		
10	I	40	F		
11	J	60	A		

37. RANK

The Excel RANK function returns the rank of a numeric value when compared to a list of other numeric values.

```
=RANK(number,ref,[order])
```

- *number* - The number to rank.
- *ref* - A range that contains numbers to rank against.
- *order* - [optional] Whether to rank in ascending or descending order.



	A	B	C	D	E
1	NAME	SCORE	RANK		
2	A	100	10		
3	B	50	5		
4	C	30	3		
5	D	80	8		
6	E	10	1		
7	F	90	9		
8	G	70	7		
9	H	20	2		
10	I	40	4		
11	J	60	6		

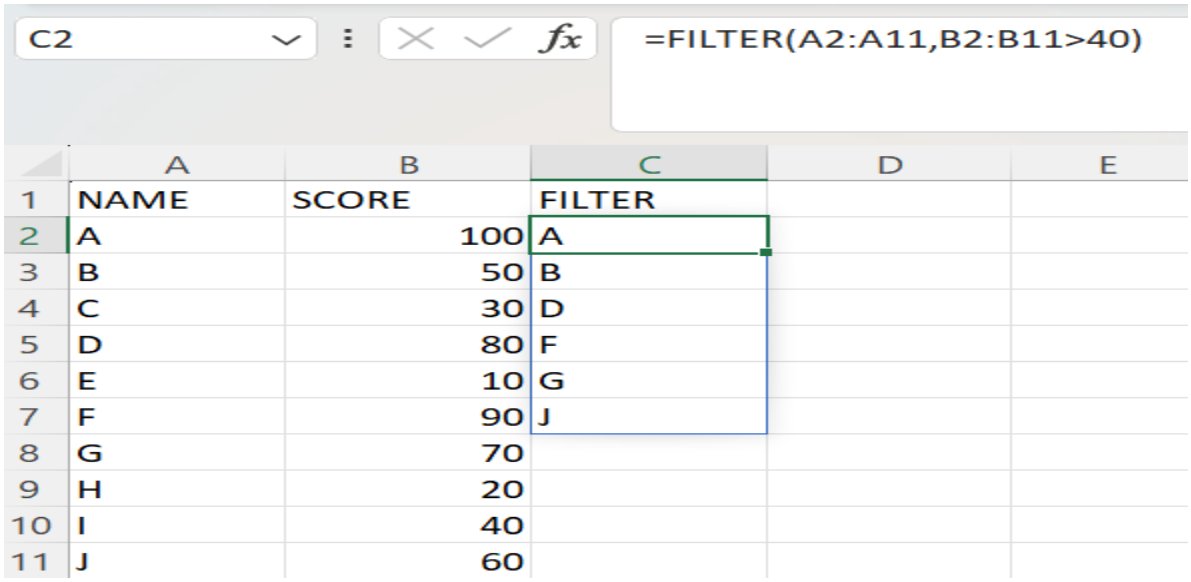
38. FILTER

The Excel FILTER function is used to extract matching values from data based on one or more conditions.

Syntax

```
=FILTER(array,include,[if_empty])
```

- *array* - Range or array to filter.
- *include* - Boolean array, supplied as criteria.
- *if_empty* - [optional] Value to return when no results are returned.



	A	B	C	D	E
1	NAME	SCORE	FILTER		
2	A	100	A		
3	B	50	B		
4	C	30	D		
5	D	80	F		
6	E	10	G		
7	F	90	J		
8	G	70			
9	H	20			
10	I	40			
11	J	60			

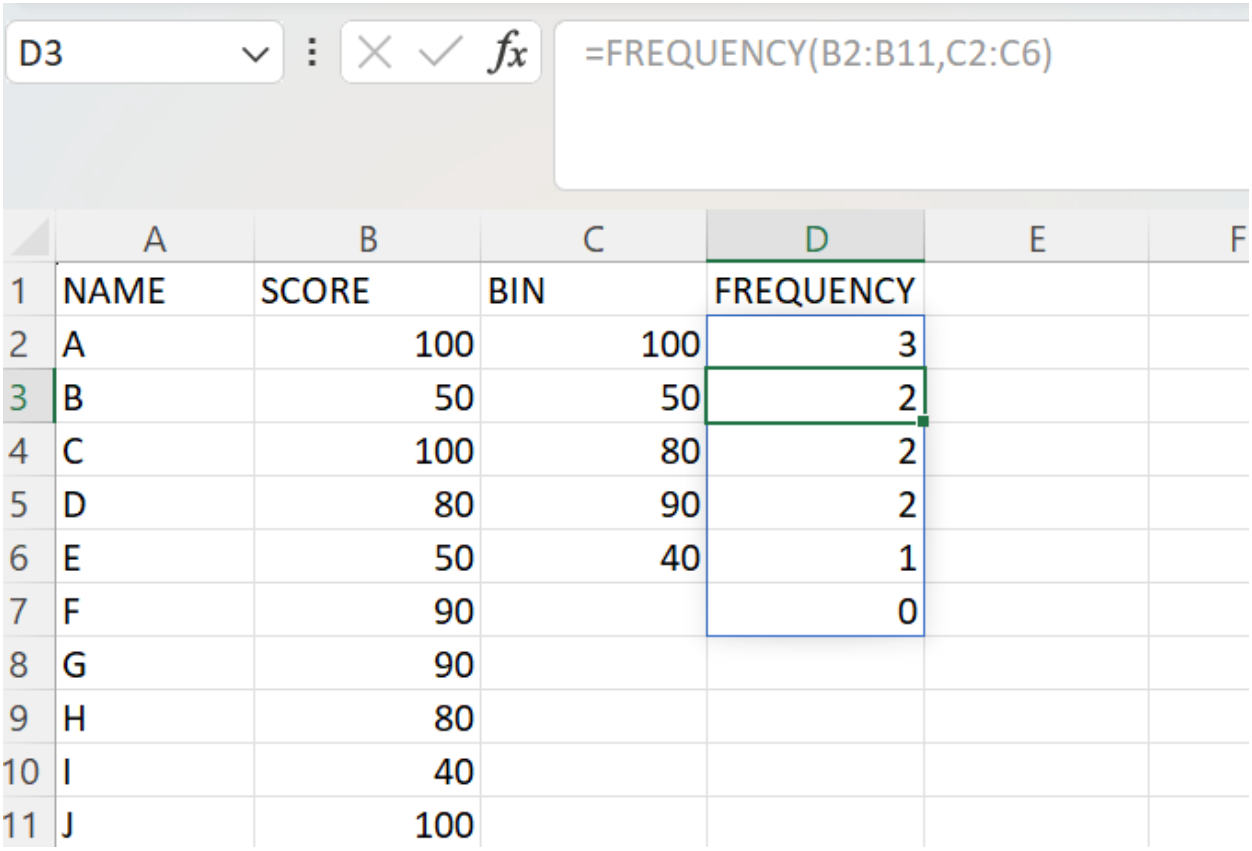
39. FREQUENCY

The Excel FREQUENCY function returns a frequency distribution, which is a list that shows the frequency of values at given intervals.

Syntax

```
=FREQUENCY(data_array,bins_array)
```

- *data_array* - An array of values for which you want to get frequencies.
- *bins_array* - An array of intervals ("bins") for grouping values.



	A	B	C	D	E	F
1	NAME	SCORE	BIN	FREQUENCY		
2	A	100	100	3		
3	B	50	50	2		
4	C	100	80	2		
5	D	80	90	2		
6	E	50	40	1		
7	F	90		0		
8	G	90				
9	H	80				
10	I	40				
11	J	100				

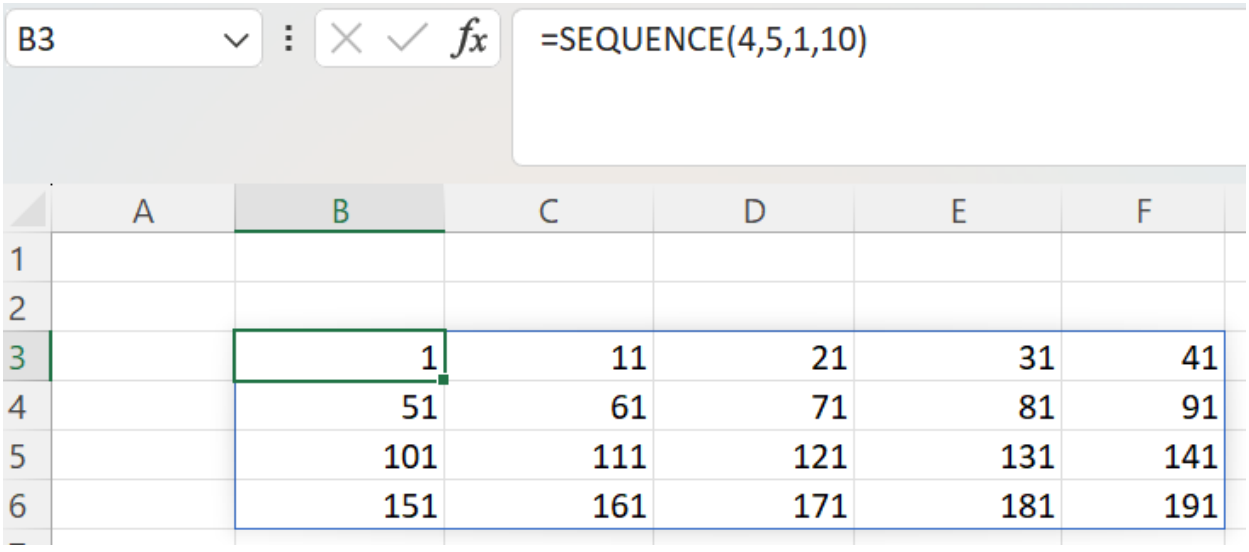
40. SEQUENCE

The Excel SEQUENCE function generates a list of sequential numbers in an array.

Syntax

```
=SEQUENCE(rows,[columns],[start],[step])
```

- *rows* - Number of rows to return.
- *columns* - [optional] Number of columns to return.
- *start* - [optional] Starting value (defaults to 1).
- *step* - [optional] Increment between each value (defaults to 1).



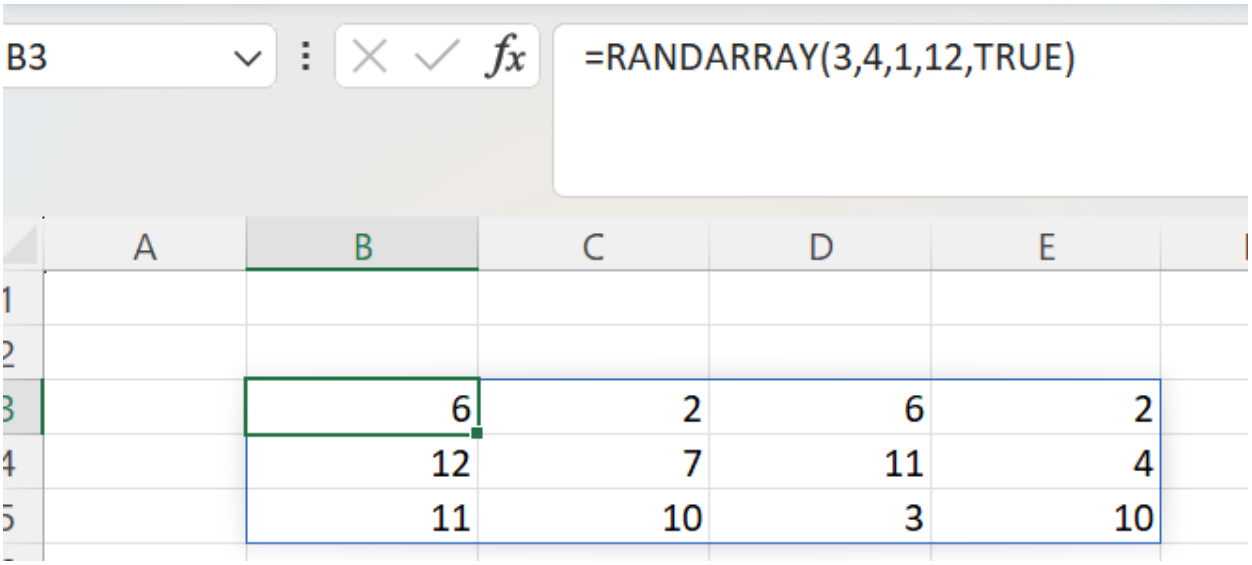
41. RANDARRAY

The Excel RANDARRAY function generates an array of random numbers between two values. The size or the array is specified by *rows* and *columns* arguments. The generated values can be either decimals or whole numbers.

Syntax

```
=RANDARRAY([rows],[columns],[min],[max],[integer])
```

- *rows* - [optional] Row count. Default = 1.
- *columns* - [optional] Column count. Default = 1.
- *min* - [optional] Minimum value. Default = 0.
- *max* - [optional] Maximum value. Default = 1.
- *integer* - [optional] Whole numbers. Boolean, TRUE or FALSE. Default = FALSE.



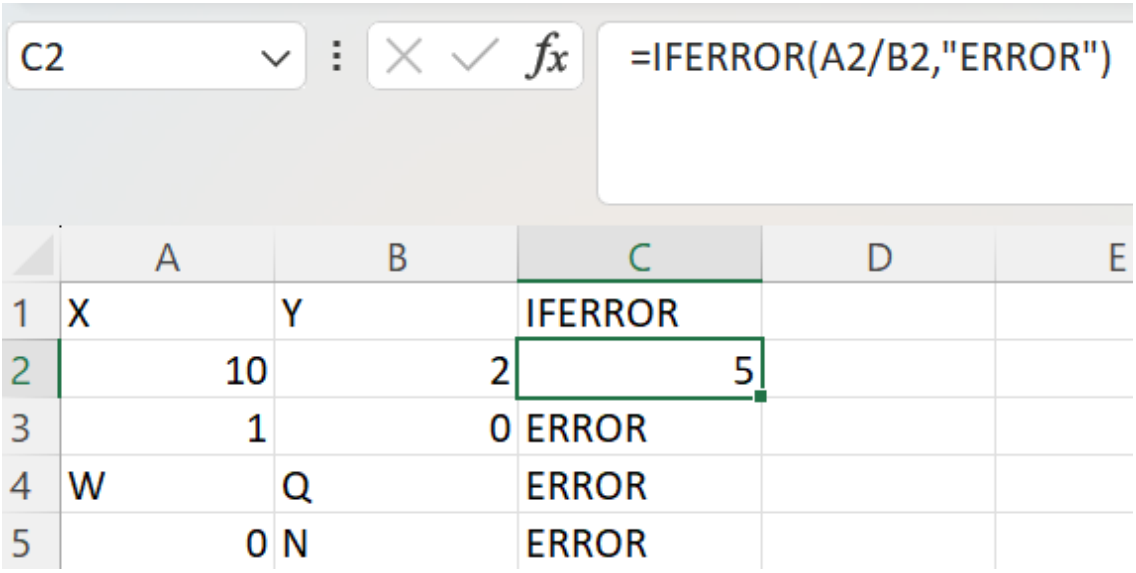
42. IFERROR

The Excel IFERROR function returns a custom result when a formula generates an error, and a standard result when no error is detected.

Syntax

```
=IFERROR(value,value_if_error)
```

- *value* - The value, reference, or formula to check for an error.
- *value_if_error* - The value to return if an error is found.



	A	B	C	D	E
1	X	Y	IFERROR		
2	10	2	5		
3	1	0	ERROR		
4	W	Q	ERROR		
5	0	N	ERROR		

PIVOT TABLES, WHAT-IF ANALYSIS, DATA VALIDATION, SUBTOTALS WITH RANGES

PIVOT TABLES:

A PivotTable is a powerful tool to calculate, summarize, and analyze data that lets you see comparisons, patterns, and trends in your data. PivotTables work a little bit differently depending on what platform you are using to run Excel.

WHAT-IF ANALYSIS :

- In what-if analysis, various situations or "scenarios" are contemplated through the manipulation of variables. Among others, there are two very popular types of what-if analysis: scenario and sensitivity analysis.
- In sensitivity analysis, the focus is on the effects of changes to one specific variable; this type of analysis explains how variables (dependent) are affected based on changes made in another variable (independent).
- In scenario analysis, the focus is on the effects of changes to multiple variables; here, the different scenarios correspond to these changing variables. For example, a worst- case scenario could involve higher-than-average costs and minimum sales.

Part- C: Power BI

1. Introduction to Power BI- Get Started with Power BI - Sign up for Power BI - Overview: Power BI data sources - Connect to a SaaS solution - Upload a local CSV file - Connect to Excel data that can be refreshed - Create a Report with Visualizations

1. Introduction to Power BI

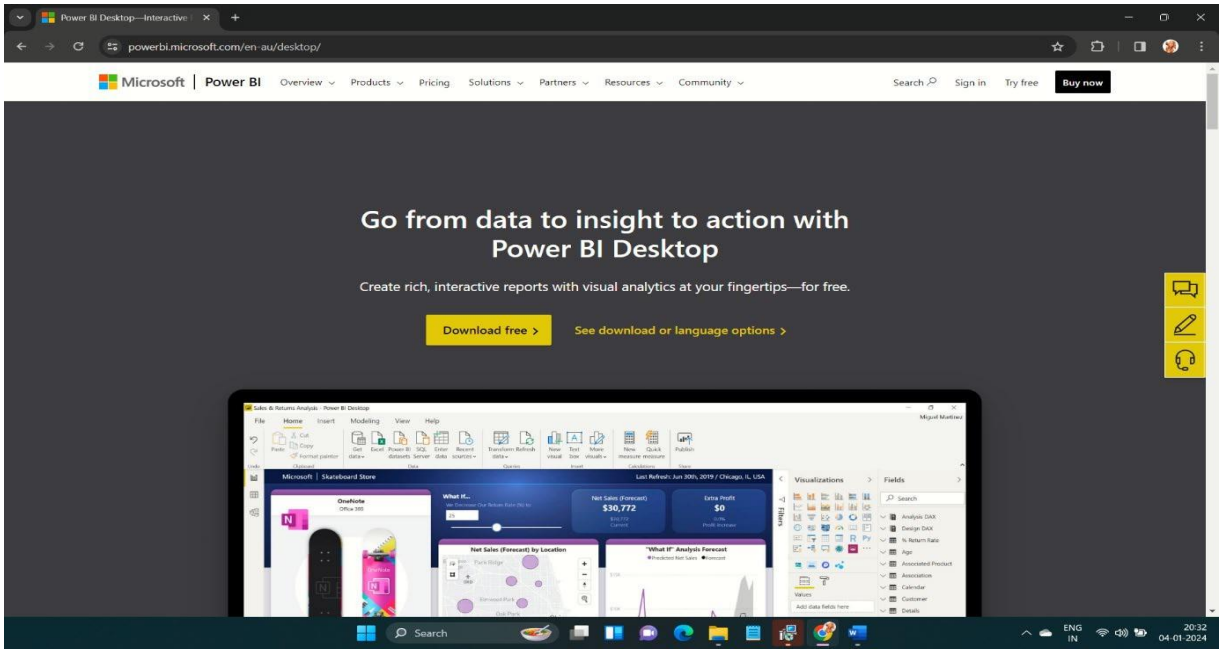
Power BI is a powerful business analytics tool developed by Microsoft that enables users to visualize and analyze data effectively, helping organizations make data-driven decisions.

Power BI Desktop:

This is a free application for Windows that allows users to create reports and data visualizations. It includes tools for data modelling, transforming data using Power Query, creating relationships, and designing interactive reports and dashboards.

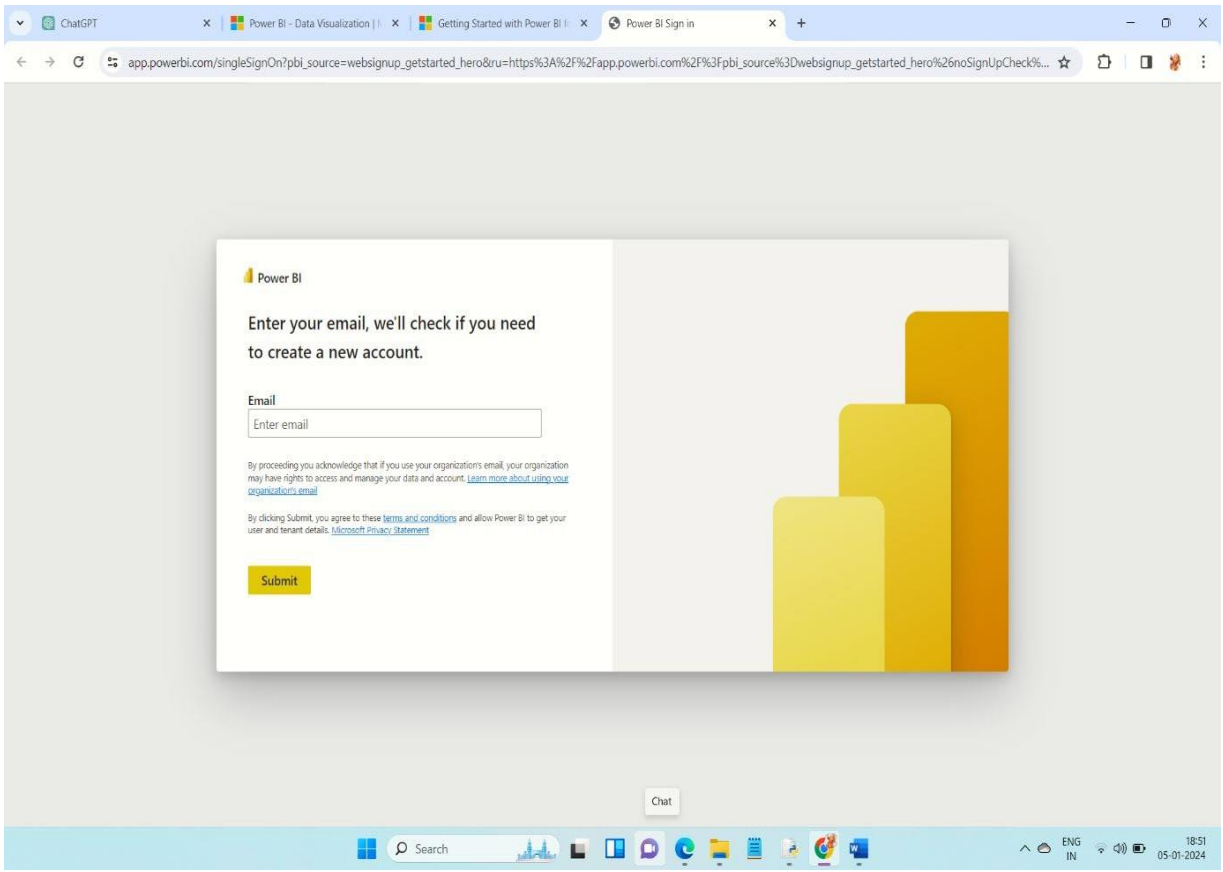
2. Get Started with Power BI

- Go to the official Power BI Desktop download page (https://powerbi.microsoft.com/en-us/desktop/).
- Download Power BI by clicking on the button indicating "Download free".
- Once the download is complete, follow the on-screen instructions provided by the installation wizard to install Power BI Desktop on your computer.
- After successful installation, you can launch Power BI Desktop.



3. Sign up for Power BI:

- Go to the Power BI website (powerbi.microsoft.com).
- Sign up for a Power BI account using your work or personal email.



4. Overview: Power BI Data Sources:

Power BI supports various data sources, including but not limited to:

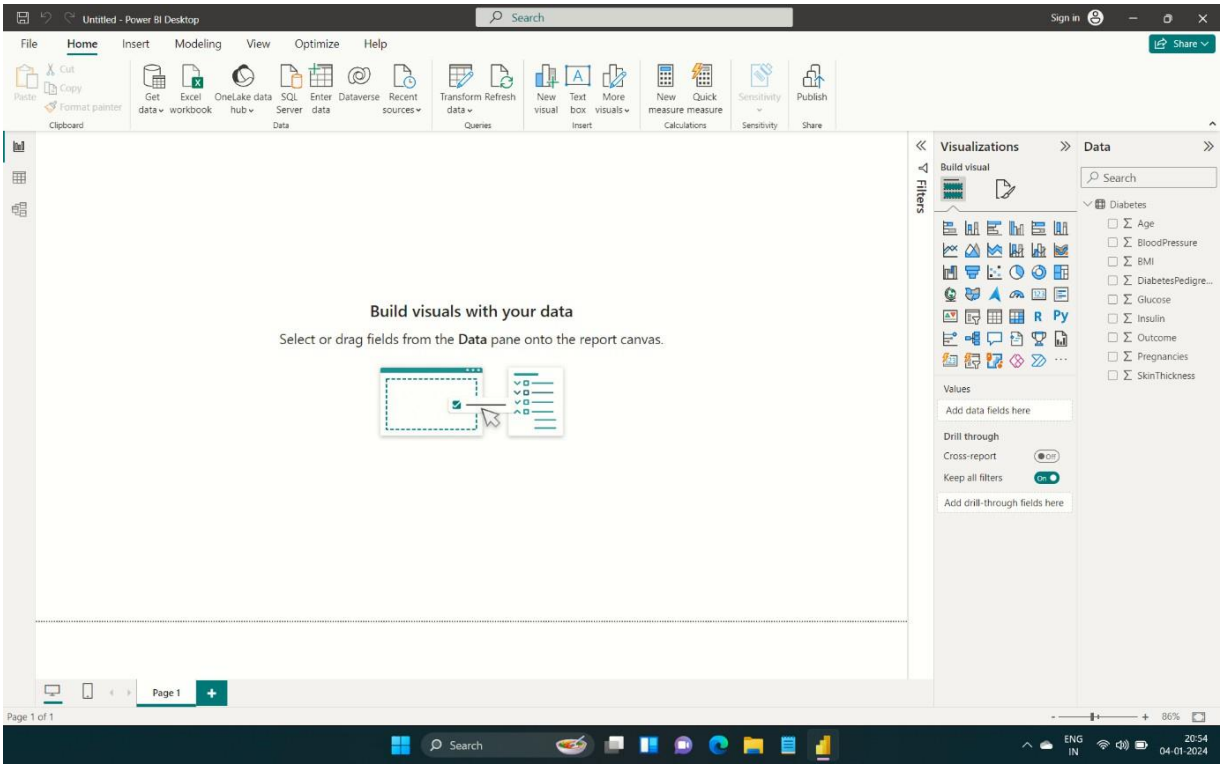
- **SaaS Solutions:** Connect to cloud-based services like Microsoft Dynamics 365, Google Analytics, Salesforce, etc.
- **Local Files:** Upload files from your computer, such as CSV, Excel, JSON, etc.
- **Databases:** Connect to databases like SQL Server, MySQL, PostgreSQL, etc.
- **Web Sources:** Extract data from web pages or APIs.
- **Azure Services:** Utilize Azure data services like Azure SQL Database, Azure Blob Storage, etc.

5. Connect to a SaaS Solution:

- Open Power BI Desktop or sign in to the Power BI service online.
- Click on "Get Data" from the Home tab.
- Choose the type of SaaS solution you want to connect to (e.g., Salesforce, Google Analytics).
- Follow the prompts to sign in and connect to your SaaS solution. You'll likely need to provide credentials or permissions to access the data.

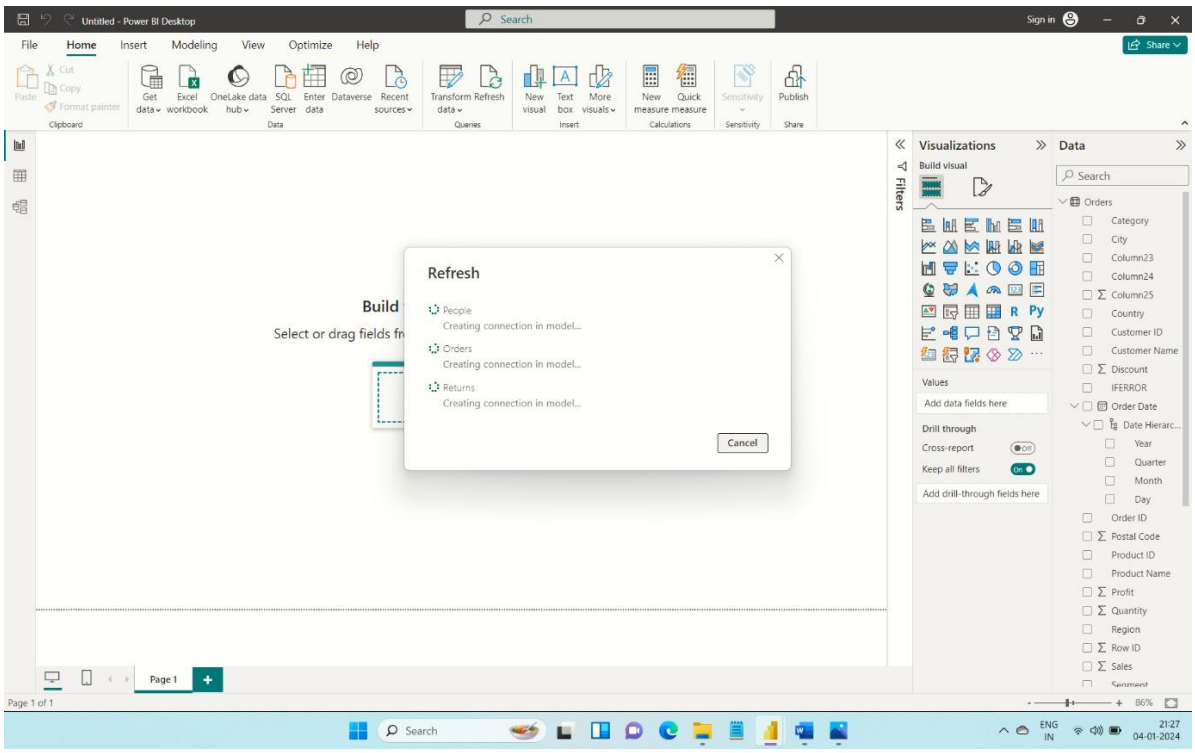
6. Upload a local CSV file:

- Click on "Get Data" from the Home tab.
- Select "Text/CSV" from the available data sources.
- Browse to locate and select the CSV file from your computer.
- Power BI will import the data from the CSV file.



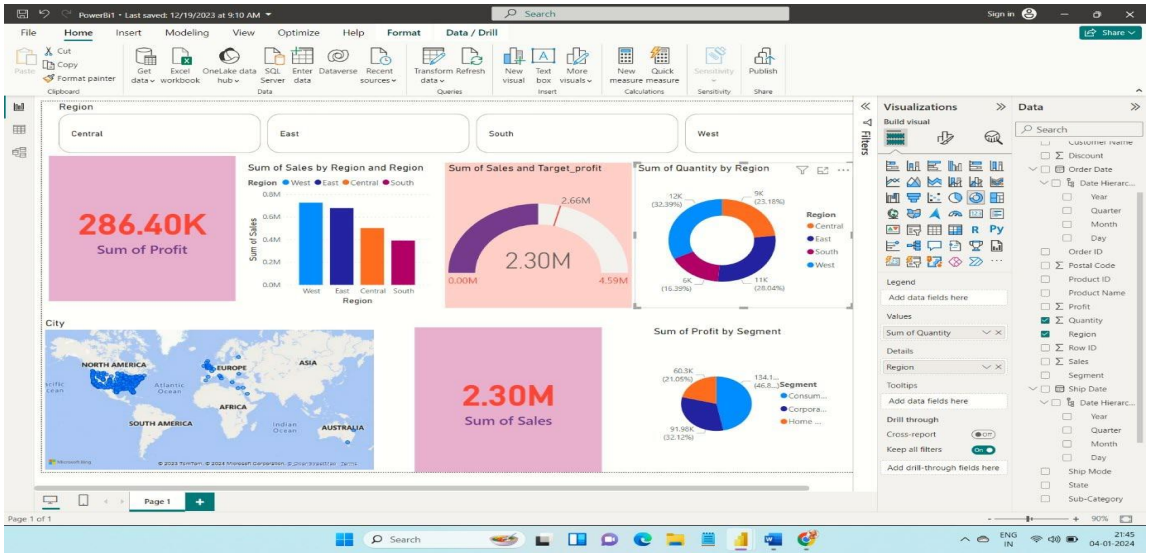
7. Connect to Excel that can be refreshed:

- Click on "Get Data" from the Home tab.
- Choose "Excel" as the data source.
- Browse to select the Excel file or specify the file path.
- Power BI will import data from the Excel file.
- To refresh the data, go to "Home" > "Queries" > "Refresh."



8. Create a Report with Visualizations:

- Select the appropriate type of visualization (e.g., bar chart, line chart, pie chart) from the Visualizations pane on the right-hand side.
- Drag and drop fields from your datasets onto the report canvas to create visualizations such as charts, graphs, tables, etc.
- Customize the visualizations by formatting, sorting, and adding additional elements like slicers, text boxes, etc.
- Repeat the process to create multiple visualizations for various insights.
- Save your report.

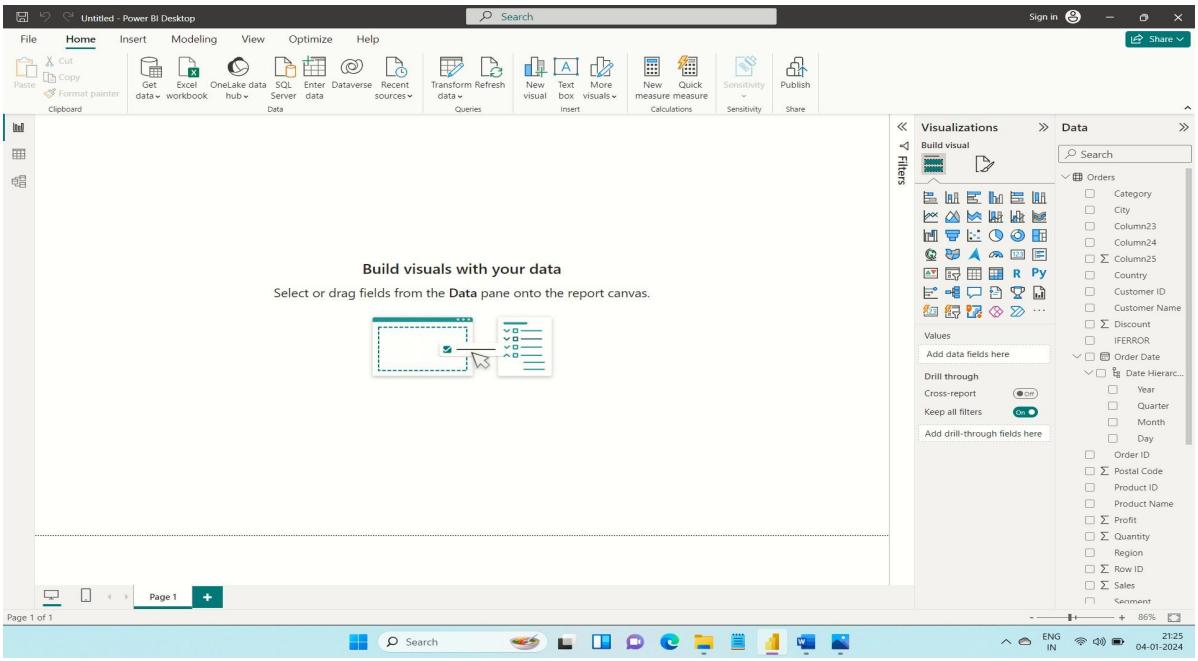


2. Using visualizations - Create a new report - Create and arrange visualizations - Format a visualization - Use text, map, and gauge visualizations and save a report - Use a slicer to filter visualizations - Sort, copy, and paste visualizations

1. Create a New Report:

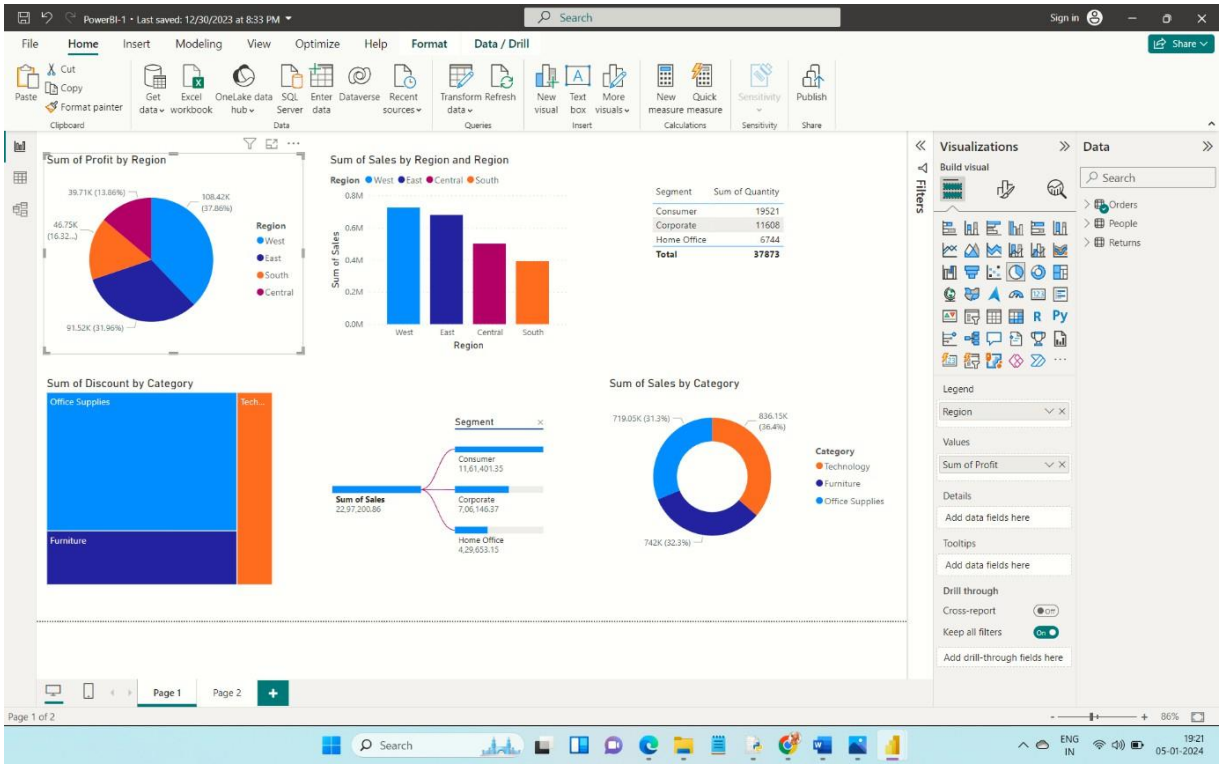
- Open Microsoft Power BI Desktop.
- Click on "File" > "New"

- Import your data into Power BI by connecting to a data source.



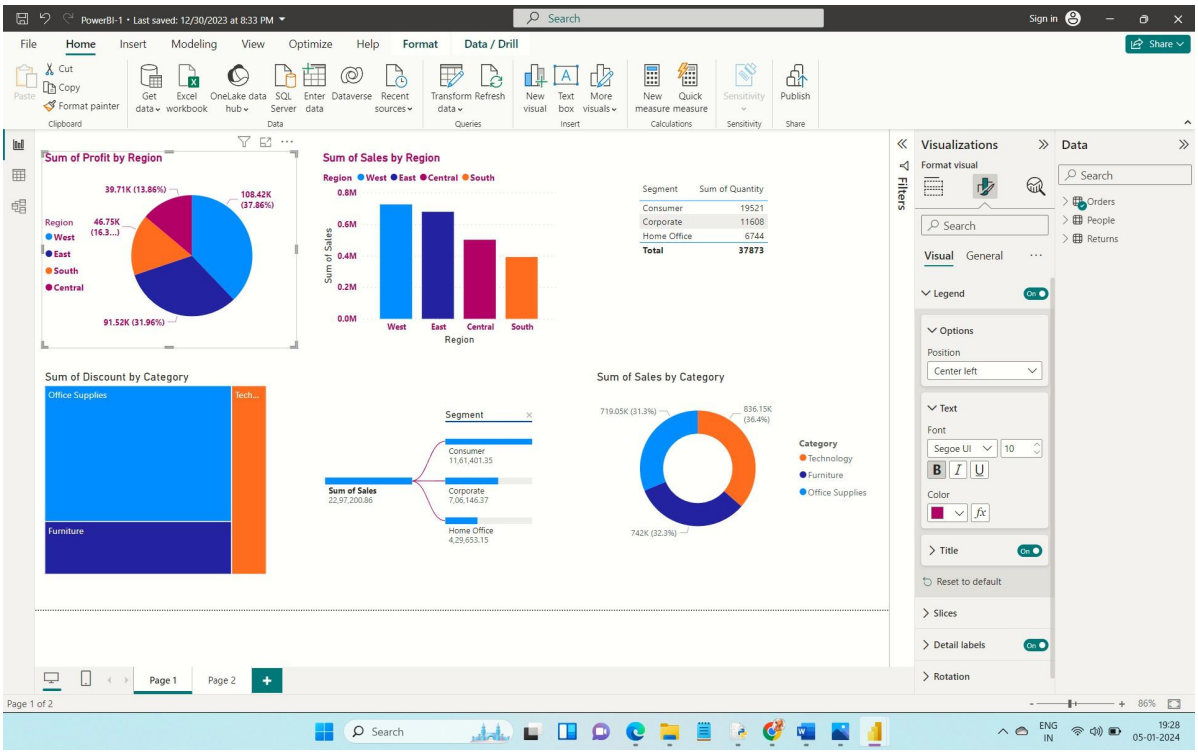
2. Create and Arrange Visualizations:

- Select the appropriate type of visualization (e.g., bar chart, line chart, pie chart) from the Visualizations pane on the right-hand side.
- Drag and drop fields from your dataset onto the visual to create different visualizations like bar charts, line graphs, etc.
- Arrange the visualizations by clicking and dragging them to desired locations on the canvas.



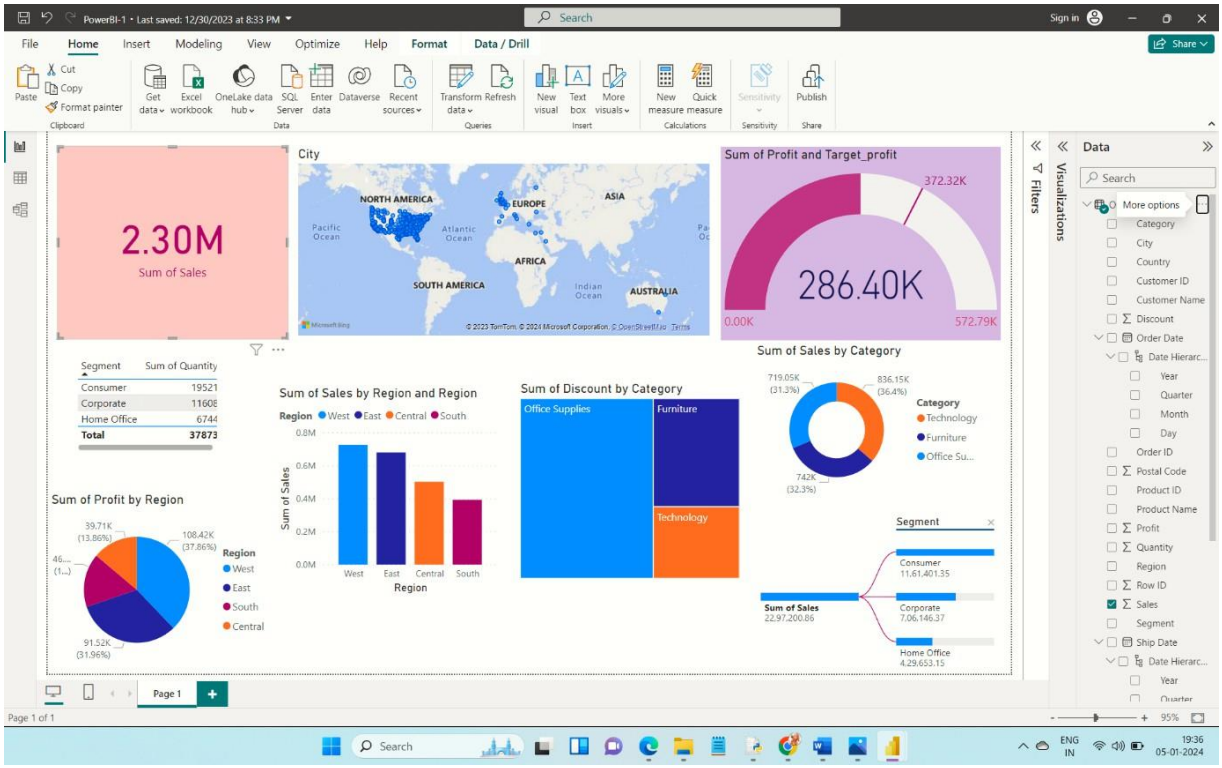
3. Format a Visualization:

- Click on a visualization to select it.
- Use the "Format" or "Visualizations" pane to modify the appearance, colors, labels, and other settings of the selected visualization.



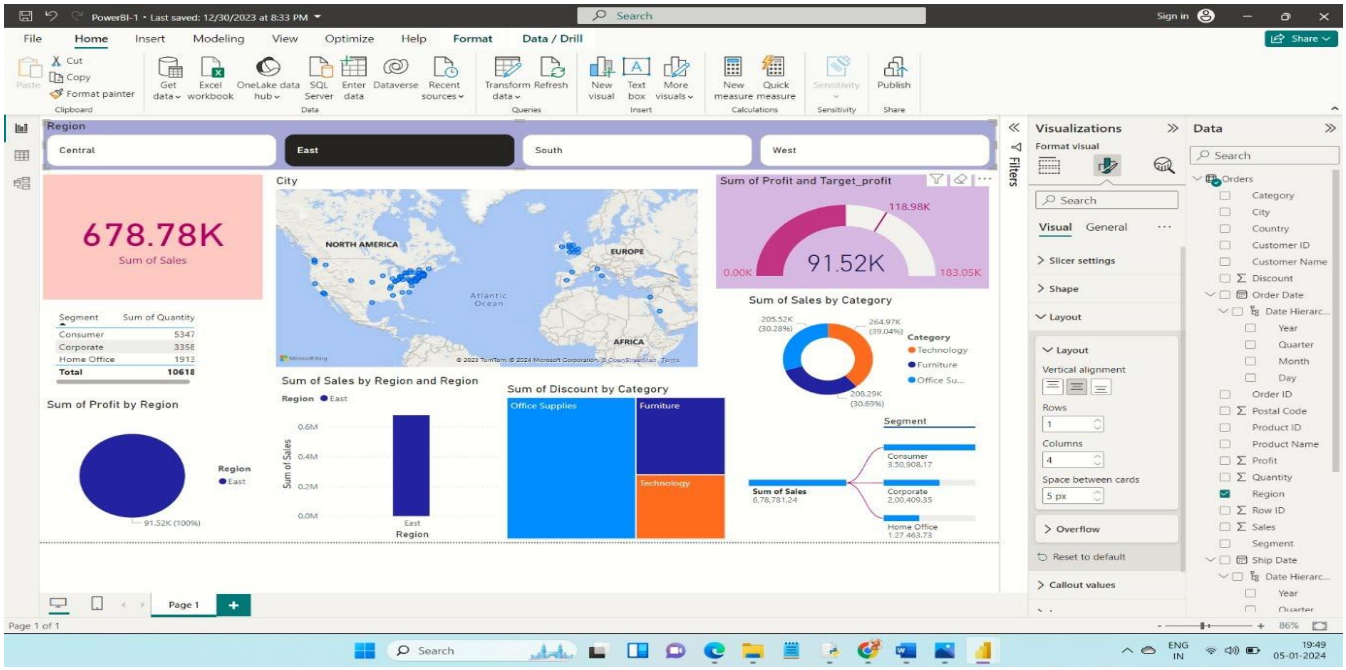
4.Use text, map, and gauge visualizations and save a report:

- Include a text visualization by selecting "Card" from the "Visualizations" pane and drag and drop fields from your dataset onto the visual
- Add a map visualization using the map visual from the visualization pane and dragging a geographical field onto the visual.
- Include a gauge visualization by selecting "Gauge" from the "Visualizations" pane and configuring it with appropriate data.
- Click on "File" > "Save" to save the report in your desired location and format.



6. Use a Slicer to Filter Visualizations:

- Select "Slicer (new)" from the Visualization pane and drag a field you want to use as a filter into the slicer visual.
- Use the slicer to interactively filter the other visualizations on the report by selecting specific values.



7. Sort, Copy, and Paste Visualizations:

- To sort visualizations, select a visualization and use the sort options available in the "Format" or "Visualizations" pane.
- To copy and paste visualizations, select the visualization, right-click, and choose "Copy." Then, right-click on the canvas and select "Paste."

4. Modify and Print a Report - Rename and delete report pages - Add a filter to a page or report Set visualization interactions - Send a report to PowerPoint

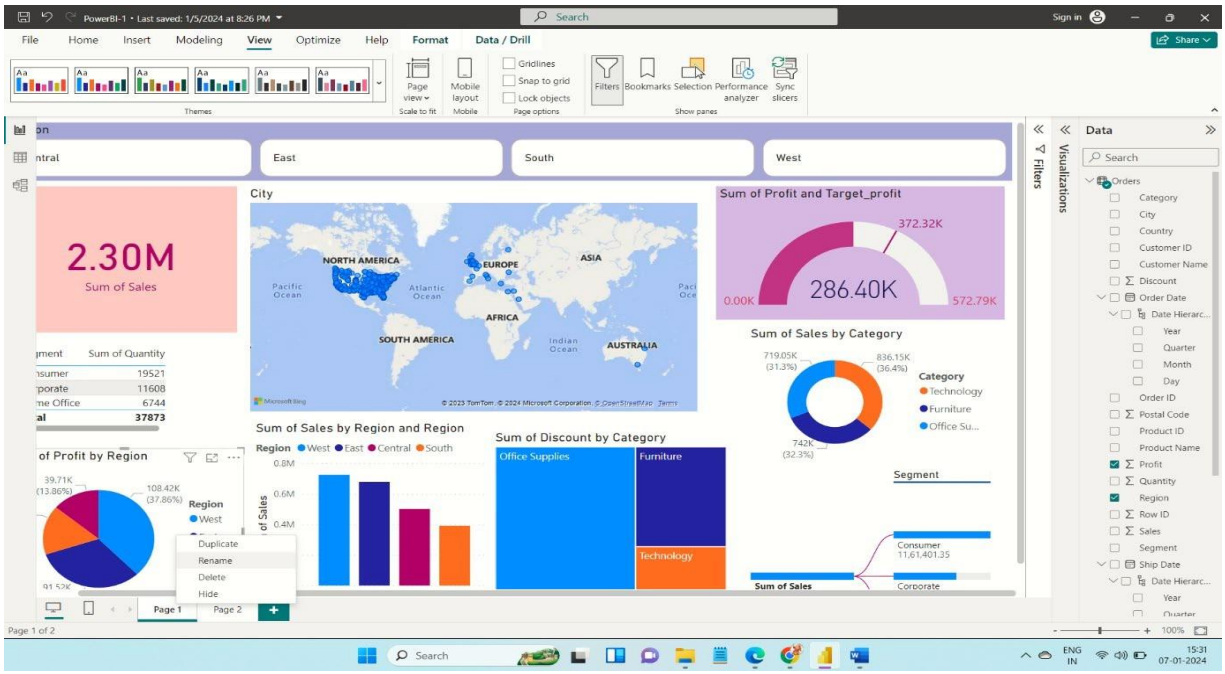
1. Modify and Print a Report:

To modify a report in Power BI, you can edit existing visuals, add new visuals, change formatting, and adjust data connections.

- Open your report in Power BI Desktop or Power BI Service.
- Make the necessary modifications to the visuals by editing their properties, formatting, or data.
- To print the report, click on "File" and then select "Print" to configure the printing settings and print the report.

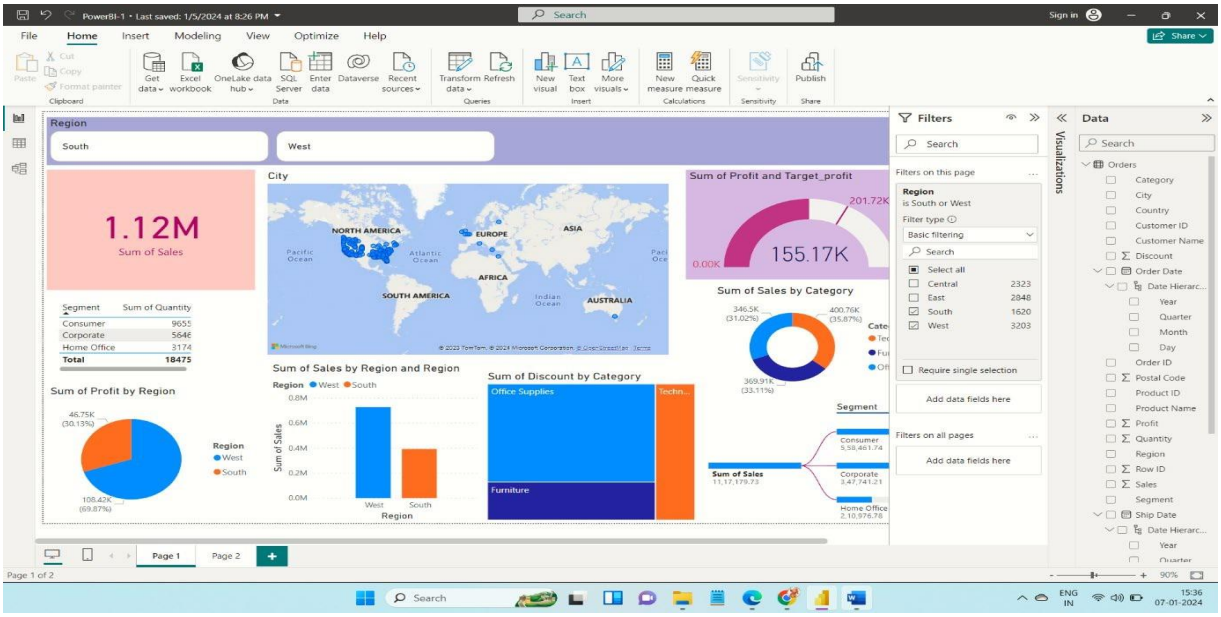
2. Rename and Delete Report Pages:

- In Power BI Desktop or Service, navigate to the report page.
- To rename a report page, right-click on the page name in the Pages pane, select "Rename," and enter the new name.
- To delete a report page, right-click on the page name in the Pages pane and select "Delete."



1. Add a Filter to a Page or Report:

- In Power BI Desktop, select the page or visual you want to apply the filter to.
- Go to the "Filters" pane and click on "Add a filter."
- Choose the field you want to use as a filter and configure its settings (e.g., filter type, values).



4.Set Visualization Interactions:

- Click on the visual you want to set interactions for.
- Go to the "Format" pane (or right-click and select "Format") and navigate to the "Edit interactions" option.
- Adjust the interactions between visuals by selecting the interaction type (e.g., highlighting, filtering) for each visual.

3. Send a Report to PowerPoint:

- In Power BI Service, open the report you want to export.
- Click on "File" and select "Export" > "PowerPoint (.pptx)."
- Configure the export settings and click "Export" to generate the PowerPoint file with the report visuals.

PART B

1. A) Calculating the simple probabilities

```
import random
# Taking user inputs
num_trials = int(input("Enter number of trials: "))
rolls_per_trial = int(input("For each trial, how many rolls: "))
roll_up_value = int(input("Enter the roll-up value (e.g., the number you want to count): "))

poss_outcomes = 0

# Rolling the dice for the specified number of trials and rolls
for i in range(num_trials):
    for j in range(rolls_per_trial):
        result = random.randint(1, 6)
        print(result, end=" ")  # Printing each roll result on the same line
        if result == roll_up_value:
            poss_outcomes += 1
    print()  # Adding a newline after each trial

# Calculating total outcomes and probability
total_outcomes = num_trials * rolls_per_trial
print(f"Number of times {roll_up_value} appeared in {num_trials} trials of {rolls_per_trial} rolls each:
    {poss_outcomes}")
print(f"Probability = {poss_outcomes / total_outcomes}")
```

```
Enter number of trials: 5
For each trial, how many rolls: 2
Enter the roll-up value (e.g., the number you want to count): 6
6 6
3 6
2 3
4 4
5 1
Number of times 6 appeared in 5 trials of 2 rolls each: 3
Probability = 0.3
```

1. b) Applications of Probability distributions to real life problems

```
from scipy.stats import binom

n = 10  # Number of trials
p = 0.5  # Probability of success
k_success = 2  # Number of successes

# Calculating the probability of exactly 2 successes
prob_2_success = binom.pmf(k_success, n, p)

print(f"Probability of 2 successes out of 10 trials: {prob_2_success}")
```

output:

Probability of 2 successes out of 10 trials: 0.04394531250000005

2. Test of significance

```
import pandas as pd
from scipy import stats
titanic_data = pd.read_csv(r"C:\Users\Rohit\Desktop\train.csv")
# One Sample T-Test: Checking mean age against a hypothetical mean
hypothetical_mean_age = 30
ttest_one_sample = stats.ttest_1samp(titanic_data['Age'].dropna(),
hypothetical_mean_age)
print("One Sample T-Test:")
print("T-statistic:", ttest_one_sample.statistic)
print("p-value:", ttest_one_sample.pvalue)

        # Two Independent Samples T-Test: Comparing ages of male and female passengers

male_ages = titanic_data[titanic_data['Sex'] == 'male']['Age'].dropna()
female_ages = titanic_data[titanic_data['Sex'] == 'female']['Age'].dropna()
ttest_two_ind_samples = stats.ttest_ind(male_ages, female_ages)
print("\nTwo Independent Samples T-Test:")
print("T-statistic:", ttest_two_ind_samples.statistic)
print("p-value:", ttest_two_ind_samples.pvalue)
# Paired T-Test: Comparing fares before and after

before_fares = titanic_data['Fare'].dropna()
after_fares = before_fares * 1.2 # Assuming a 20% increase in fares
ttest_paired = stats.ttest_rel(before_fares, after_fares)
print("\nPaired T-Test:")
print("T-statistic:", ttest_paired.statistic)
```

```
print("p-value:", ttest_paired.pvalue)
# ANOVA Test: Impact of passenger class on fares

anova_result = stats.f_oneway(titanic_data[titanic_data['Pclass'] == 1]['Fare'].dropna(),
titanic_data[titanic_data['Pclass'] == 2]['Fare'].dropna(),
titanic_data[titanic_data['Pclass'] == 3]['Fare'].dropna())
print("\nANOVA Test Result:")
print("F-statistic:", anova_result.statistic)
print("p-value:", anova_result.pvalue)
# Chi-Square Test: Relationship between survival status and passenger class

chi2_table = pd.crosstab(titanic_data['Survived'], titanic_data['Pclass'])
chi2_result = stats.chi2_contingency(chi2_table)
print("\nChi-Square Test Result:")
print("Chi-Square statistic:", chi2_result[0])
print("p-value:", chi2_result[1])
```

OUTPUT:

```
One Sample T-Test:
T-statistic: -0.5534583115970276
p-value: 0.5801231230388639

Two Independent Samples T-Test:
T-statistic: 2.499206354920835
p-value: 0.012671296797013709

Paired T-Test:
T-statistic: -19.344277455944212
p-value: 7.255925461999273e-70

ANOVA Test Result:
F-statistic: 242.34415651744814
p-value: 1.0313763209141171e-84

Chi-Square Test Result:
Chi-Square statistic: 102.88898875696056
p-value: 4.549251711298793e-23
```

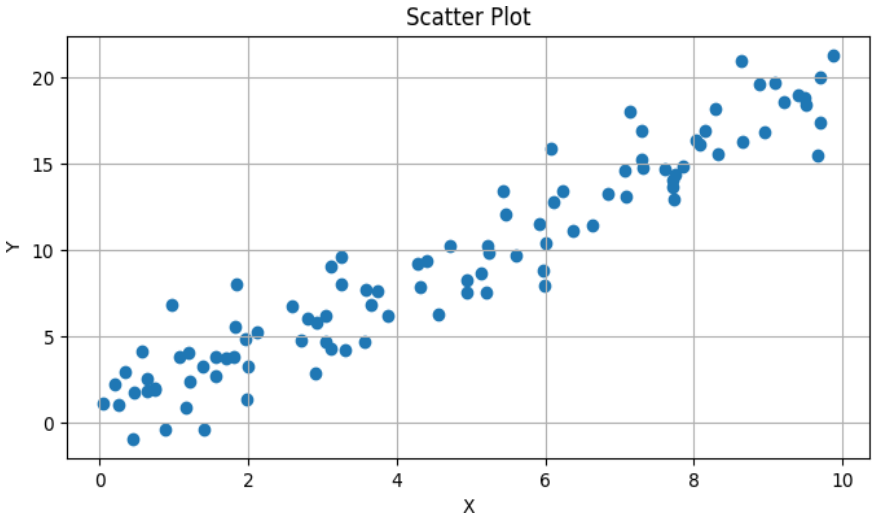
Program 3:
Correlation and Regression analysis
Scattered diagram, calculating of correlation coefficient
Linear regression: fitting, testing model adequacy and prediction
Fitting of logistic regression.

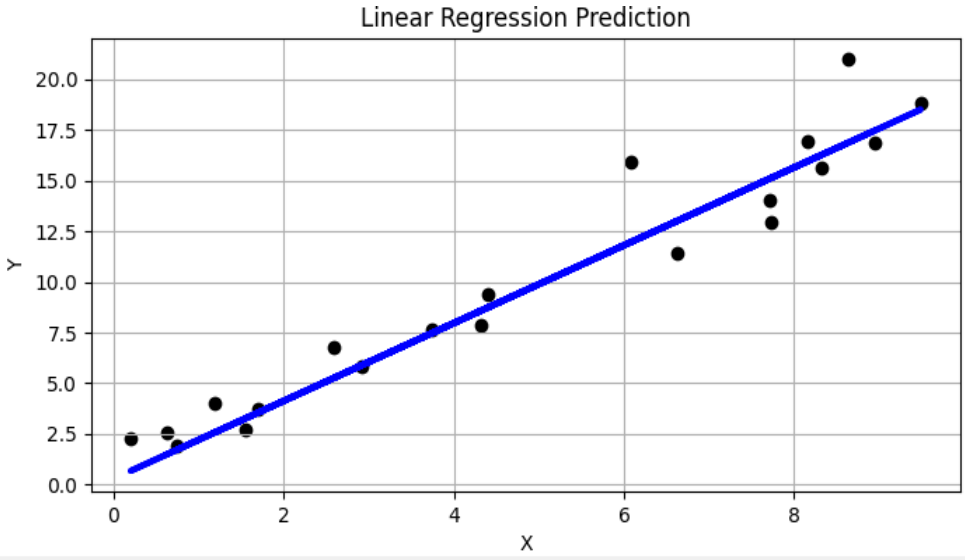
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris

# Generating sample data
np.random.seed(42)
X = np.random.rand(100, 1) * 10
y = 2 * X.squeeze() + np.random.randn(100) * 2

# Scatter plot and correlation coefficient
plt.figure(figsize=(8, 4))
plt.scatter(X, y)
plt.title('Scatter Plot')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)

correlation_coefficient = np.corrcoef(X.squeeze(), y)[0, 1]
print(f"Correlation Coefficient: {correlation_coefficient}")

# Linear regression fitting
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)

# Testing model adequacy and prediction
y_pred = lin_reg.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R-squared Score: {r2}")
```

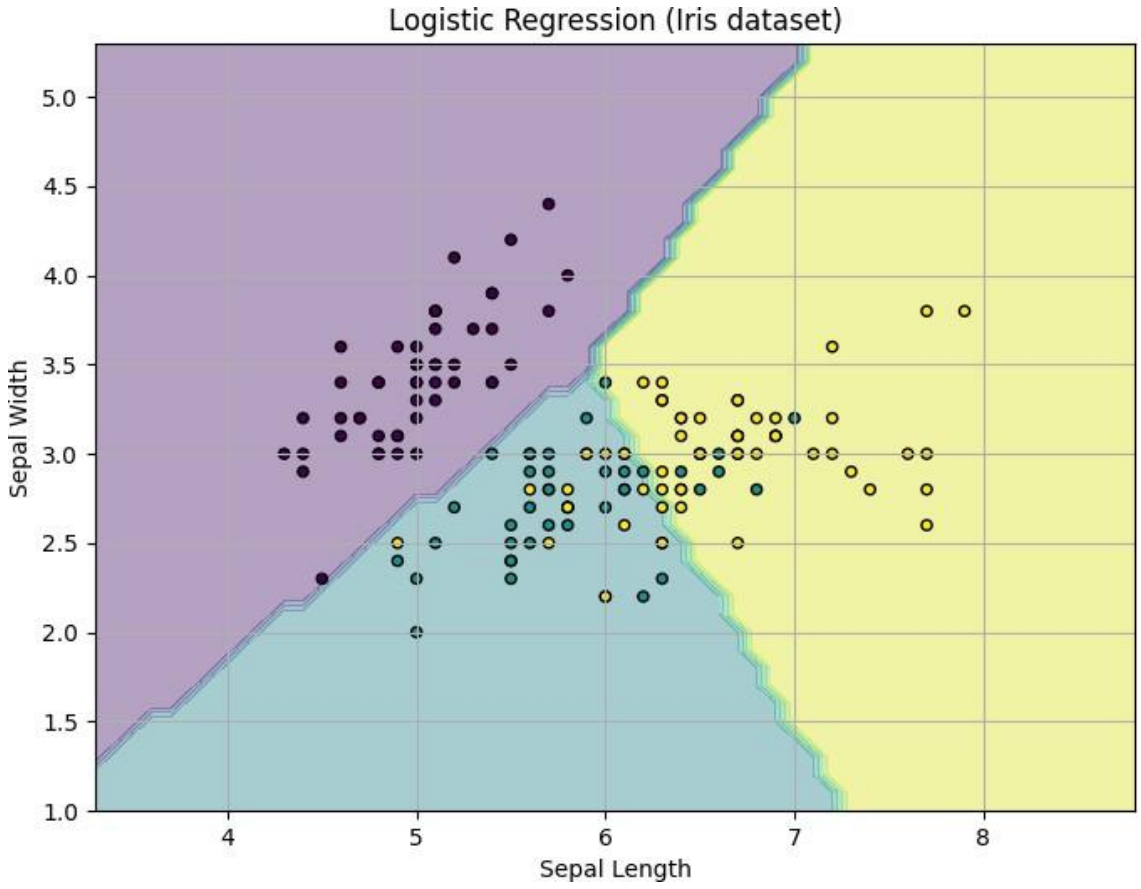
```
# Plotting linear regression prediction
plt.figure(figsize=(8, 4))
plt.scatter(X_test, y_test, color='black')
plt.plot(X_test, y_pred, color='blue', linewidth=3)
plt.title('Linear Regression Prediction')
plt.xlabel('X')
plt.ylabel('Y')
plt.grid(True)

# Fitting logistic regression (using Iris dataset as an example)
iris = load_iris()
X_iris = iris.data[:, :2]  # Using only the first two features for simplicity
y_iris = iris.target

log_reg = LogisticRegression()
log_reg.fit(X_iris, y_iris)

# Generating a meshgrid for decision boundary visualization
x_min, x_max = X_iris[:, 0].min() - 1, X_iris[:, 0].max() + 1
y_min, y_max = X_iris[:, 1].min() - 1, X_iris[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1), np.arange(y_min, y_max, 0.1))
Z = log_reg.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

# Plotting logistic regression decision boundary
plt.figure(figsize=(8, 6))
plt.contourf(xx, yy, Z, alpha=0.4)
plt.scatter(X_iris[:, 0], X_iris[:, 1], c=y_iris, s=20, edgecolor='k')
plt.title('Logistic Regression (Iris dataset)')
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')
plt.grid(True)

# Show all plots
plt.show()
```

Output:

```
Correlation Coefficient: 0.9529657473628446
Mean Squared Error: 2.6147980548680088
R-squared Score: 0.9287298556395622
```


Scatter Plot
Y
X
0
5
10
15
20
0
2
4
6
8
10


Linear Regression Prediction
Y
X
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
0
2
4
6
8


Logistic Regression (Iris dataset)
Sepal Width
Sepal Length
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
4
5
6
7
8