Data Analytics Lab

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
5. OFFSET, CHOOSE, LET, MAX, SORT, SORTBY, RANK
6. FILTER, FRQUENCY, SEQUENCE, RANDARRAY, IFERROR
7. PIVOT TABLES, WHAT-IF ANALYSIS, DATA VALIDATION, SUBTOTALS WITH RANGES
8. 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

LAB PROGRAM -1

1. CONDITIONAL FORMATTING, IF, COUNTIF, SUMIF, AVERAGE, CONCAT using Excel Formulas.

1. Apply conditional formatting to highlight employees with high salary greater than 50,000
2. Use the IF function to categorize employees into "High Performers" or "Average Performers" based on their sales.
3. Use COUNTIF, SUMIF, and AVERAGE to calculate statistics like no of employees working in each department and sum of their salaries and average of their Salaries, average Salary of total employees.
4. Create a new column using CONCAT to combine the employee's name and department.

Conditional formatting rule:

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

IF Function

The **IF** function is a premade function in Excel, which returns values based on **true** or **false condition**.It is typed =IF and has 3 parts:

=IF(**logical test**, **[value_if_true]**, **[value_if_false]**)

COUNTIF Function

The **COUNTIF** function is a predefined function in Excel, which counts cells as specified.It is typed =COUNTIF

SUMIF Function

The **SUMIF** function is a predefined function in Excel, which calculates the sum of values in a rangebased on **condition**.
It is typed =SUMIF:
Syntax:
=SUMIF (**range**, **criteria**, **[sum range]**)

The **condition** is referred to as criteria, which can check things like: >, <, >=, <=,==, !=

AVERAGE Function

The **AVERAGE** function is a premade function in Excel, which calculates the

average (arithmetic mean).It is typed =AVERAGE

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

CONCAT Function

To concatenate is to link something together.
CONCAT is a function in Excel and is short for concatenate.
The **CONCAT** function is used to link multiple cells without adding any delimiters between thecombined cell values.
It is typed =CONCAT(**cell1**, **delimiter**, **cell2**)

Example of delimiters

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

1) Conditional formatting rule:

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

1. **Apply conditional formatting to highlight employees with high salary greater than 50,000**

   HomeTab -> Conditional Formatting -> Highlight All -> Greater than -> option -> Enter the value -> Click Ok. (Salary > 50,000).

Name	Joining Date	Email	Department	Salary
Mark	31-Dec-21	Mark@demomail.com	Human Resources	50,000
Brian	31-Dec-21	Brian@demomail.com	Sales	45,000
Alan	14-Jan-22	Alan@demomail.com	Legal	25,000
Tony	14-Jan-22	Tony@demomail.com	Retail	25,000
Agatha	01-Feb-22	Agatha@demomail.com	Sales	30,000
Lana	01-Feb-22	Lana@demomail.com	Accounting	45,000
Heather	04-Aug-22	Heather@demomail.com	Accounting	50,000
Ben	04-Aug-22	Ben@demomail.com	Sales	30,000
Caitlyn	01-Mar-22	Caitlyn@demomail.com	Retail	34,000
Gibbs	01-Mar-22	Gibbs@demomail.com	Retail	25,000
Anderson	01-Mar-22	Anderson@demomail.com	Sales	24,000
Michael	01-Mar-22	Michael@demomail.com	Retail	1,00,000
David	15-Mar-22	David@demomail.com	Sales	1,25,000
Jacob	15-Mar-22	Jacob@demomail.com	Support	36,000
John	01-Apr-22	John@demomail.com	Business Development	42,000
Leonardo	01-Apr-22	Leonardo@demomail.com	Business Development	65,000
Matthew	17-Apr-22	Matthew@demomail.com	Human Resources	35,000
Joana	01-Jun-22	Joana@demomail.com	Business Development	78,000
Ross	02-Jul-22	Ross@demomail.com	Legal	23,000
Joey	15-Jul-22	Joey@demomail.com	Business Development	56,000
Jack	03-Jan-22	Jack@demomail.com	Support	45,000

2. **Use the IF function to categorize employees into "High Performers" or "Average Performers" based on their sales.**

=IF(E2>50000;"HIGH Performers";"Average_performers")

Name	Joining Date	Email	Department	Salary	IF
Mark	31-Dec-21	Mark@demomail.com	Human Resources	50,000	Average_performers
Brian	31-Dec-21	Brian@demomail.com	Sales	45,000	Average_performers
Alan	14-Jan-22	Alan@demomail.com	Legal	25,000	Average_performers
Tony	14-Jan-22	Tony@demomail.com	Retail	25,000	Average_performers
Agatha	01-Feb-22	Agatha@demomail.com	Sales	30,000	Average_performers
Lana	01-Feb-22	Lana@demomail.com	Accounting	45,000	Average_performers
Heather	04-Aug-22	Heather@demomail.com	Accounting	50,000	Average_performers
Ben	04-Aug-22	Ben@demomail.com	Sales	30,000	Average_performers
Caitlyn	01-Mar-22	Caitlyn@demomail.com	Retail	34,000	Average_performers
Gibbs	01-Mar-22	Gibbs@demomail.com	Retail	25,000	Average_performers
Anderson	01-Mar-22	Anderson@demomail.com	Sales	24,000	Average_performers
Michael	01-Mar-22	Michael@demomail.com	Retail	1,00,000	HIGH Performers
David	15-Mar-22	David@demomail.com	Sales	1,25,000	HIGH Performers
Jacob	15-Mar-22	Jacob@demomail.com	Support	36,000	Average_performers
John	01-Apr-22	John@demomail.com	Business Development	42,000	Average_performers
Leonardo	01-Apr-22	Leonardo@demomail.com	Business Development	65,000	HIGH Performers
Matthew	17-Apr-22	Matthew@demomail.com	Human Resources	35,000	Average_performers
Joana	01-Jun-22	Joana@demomail.com	Business Development	78,000	HIGH Performers
Ross	02-Jul-22	Ross@demomail.com	Legal	23,000	Average_performers
Joey	15-Jul-22	Joey@demomail.com	Business Development	56,000	HIGH Performers
Jack	03-Jan-22	Jack@demomail.com	Support	45,000	Average_performers

3. **Use COUNTIF, SUMIF, and AVERAGE to calculate statistics like no of employees working in each department and sum of their salaries, average Salary of total employees.**

Enter all the department names in a separate column called Details. Sales, Marketing, Finance, HR.

To count the number of employees working in each department.

1. To calculate the total number of employees in each department.

=COUNTIF (D$2:D$22;F$2:F$8)

2. **To calculate the sum of Salaries in Department wise**

=SUMIF (D$2:D$22;F$2:F$8;E$2:E$22)

3. **To calculate the average Salary in all department.**

=AVERAGE (E2:E22)

4. Calculate the average salaries in each department.

=AVERAGEIF (D$2:D$22;F2;E$2:E$22)

Name	Joining Date	Department	Salary	Count_if		Sum_if	AVEARGE	AVERAGE_IF
Mark	31-Dec-21	Human Resources	50,000	Human Resources	2	85000	47048	42500
Brian	31-Dec-21	Sales	45,000	Sales	5	254000		50800
Alan	14-Jan-22	Legal	25,000	Legal	2	48000		24000
Tony	14-Jan-22	Retail	25,000	Retail	4	184000		46000
Agatha	01-Feb-22	Sales	30,000	Accounting	2	95000		47500
Lana	01-Feb-22	Accounting	45,000	Support	2	81000		40500
Heather	04-Aug-22	Accounting	50,000	Business Development	4	241000		60250
Ben	04-Aug-22	Sales	30,000					
Caitlyn	01-Mar-22	Retail	34,000					
Gibbs	01-Mar-22	Retail	25,000					
Anderson	01-Mar-22	Sales	24,000					
Michael	01-Mar-22	Retail	1,00,000					
David	15-Mar-22	Sales	1,25,000					
Jacob	15-Mar-22	Support	36,000					
John	01-Apr-22	Business Development	42,000					
Leonardo	01-Apr-22	Business Development	65,000					
Matthew	17-Apr-22	Human Resources	35,000					
Joana	01-Jun-22	Business Development	78,000					
Ross	02-Jul-22	Legal	23,000					
Joey	15-Jul-22	Business Development	56,000					
Jack	03-Jan-22	Support	45,000					

5. Create a new column using CONCAT to combine the employee's name and SALARY

=CONCATENATE (A2;"---"; E2)

Name	Joining Date	Email	Department	Salary	CONCATENATION
Mark	31-Dec-21	Mark@demomail.com	Human Resources	50,000	Mark--50000
Brian	31-Dec-21	Brian@demomail.com	Sales	45,000	Brian--45000
Alan	14-Jan-22	Alan@demomail.com	Legal	25,000	Alan--25000
Tony	14-Jan-22	Tony@demomail.com	Retail	25,000	Tony--25000
Agatha	01-Feb-22	Agatha@demomail.com	Sales	30,000	Agatha--30000
Lana	01-Feb-22	Lana@demomail.com	Accounting	45,000	Lana--45000
Heather	04-Aug-22	Heather@demomail.com	Accounting	50,000	Heather--50000
Ben	04-Aug-22	Ben@demomail.com	Sales	30,000	Ben--30000
Caitlyn	01-Mar-22	Caitlyn@demomail.com	Retail	34,000	Caitlyn--34000
Gibbs	01-Mar-22	Gibbs@demomail.com	Retail	25,000	Gibbs--25000
Anderson	01-Mar-22	Anderson@demomail.com	Sales	24,000	Anderson--24000
Michael	01-Mar-22	Michael@demomail.com	Retail	1,00,000	Michael--100000
David	15-Mar-22	David@demomail.com	Sales	1,25,000	David--125000
Jacob	15-Mar-22	Jacob@demomail.com	Support	36,000	Jacob--36000
John	01-Apr-22	John@demomail.com	Business Development	42,000	John--42000
Leonardo	01-Apr-22	Leonardo@demomail.com	Business Development	65,000	Leonardo--65000
Matthew	17-Apr-22	Matthew@demomail.com	Human Resources	35,000	Matthew--35000
Joana	01-Jun-22	Joana@demomail.com	Business Development	78,000	Joana--78000
Ross	02-Jul-22	Ross@demomail.com	Legal	23,000	Ross--23000
Joey	15-Jul-22	Joey@demomail.com	Business Development	56,000	Joey--56000
Jack	03-Jan-22	Jack@demomail.com	Support	45,000	Jack--45000

LAB PROGRAM 2

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

- **Calculate Value at the intersection of the 2nd row and 2nd column in the range A3:D12 & Value at the intersection of the 3rd row and 4th column in the range A3:D12 &use Index and match function to retrieve the sales of the specific employee based upon their name.**
- **Match function is used to find out Order value is present in the table or not**
- **Using Ifs Statement to Categorize the Sales Performance**
- **Use Unique function to create the List Unique regions**
- **Use COUNTIFS, SUMIFS, AverageIFS, functions to calculate statistics based on multiple Criteria, such as region, orders (>=50), sales.**

INDEX:

The INDEX function returns a value or the reference to a value from within a table or range.

Syntax: **INDEX (array, row_num, [column_num])**

The array form of the INDEX function has the following arguments:

- **Array** Required. A range of cells or an array constant.
- If array contains only one row or column, the corresponding row_num or column_num argument is optional.
- If array has more than one row and more than one column, and only row_num or column_num is used, INDEX returns an array of the entire row or column in array.
- **row_num** Required, unless column_num is present. Selects the row in array from which to return a value. If row_num is omitted, column_num is required.
- **column_num** Optional. Selects the column in array from which to return a value. If column_num is omitted, row_num is required.

1. **Calculate Value at the intersection of the 2nd row and 2nd column in the range A3:D12 & Value at the intersection of the 3rd row and 4th column in the range A3:D12 &use Index and match function to retrieve the sales of the specific employee based upon their name.**

   - =INDEX(A3:D12;2;2)
   - =INDEX(A3:D12;3;4)
   - =INDEX(D3:D12;MATCH("Lana";A3:A12;0))

Data					
Name	**Region**	**Orders**	**Sales**	**INDEX**	
Mark	East	12	65000	West	Value at the intersection of the 2nd row and 2nd column in the range A3:D12
Brian	West	45	63000	61000	Value at the intersection of the 3rd row and 4th column in the range A3:D12
Alan	South	67	61000		
Tony	North	89	56000		
Agatha	East	100	54000	43000	use Index and match function to retrieve the sales of the specific employee based upon their name.
Lana	North	125	43000		
Heather	East	140	40000		
Ben	East	150	35000		
Caitlyn	South	178	32000		
Gibbs	East	190	31000		

MATCH function:

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_array, [match_type])

- **lookup_value** Required. The value that you want to match in ***lookup_array***.
- **lookup_array** Required. The range of cells being searched.
- **match_type** Optional. The number -1, 0, or 1.

Match_type	DESCRIPTION
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 examp 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 exam TRUE, FALSE, Z-A, ...2, 1, 0, -1, -2, ..., and so on.

2. **Match is used to find out Order value is present in the table or not**
   - =MATCH(99;C3:C12;1)
   - =MATCH(140;C3:C12;0)
   - =MATCH(33000;D3:D12;-1)

Data				Match	
Name	Region	Orders	Sales		
Mark	East	12	65000		MATCH finds the largest value that is less than or equal to lookup_value. The values in
Brian	West	45	63000	4	the lookup_array argument must be placed in ascending order.
Alan	South	67	61000		**MATCH** finds the first value that is exactly equal to ***lookup_value***. The values in
Tony	North	89	56000	7	the ***lookup_array*** argument can be in any order.
Agatha	East	100	54000		MATCH finds the smallest value that is greater than or equal to lookup_value. The
Lana	North	125	43000	8	values in the lookup_array argument must be placed in descending order.
Heather	East	140	40000		
Ben	East	150	35000		
Caitlyn	South	178	32000		
Gibbs	East	190	31000		

IFS Function:

- Use the IFS function to check 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 nestedIF statements, and is much easier to read with multiple conditions.

Syntax:

= IFS(logical_test1, Value1 [logical_test2, Value2] ..., [logical_test127, Value127])

3. **Using Ifs Statement to Categorize the Sales Performance**

   =IFS (C3>90, "A", C3>80, "B", C3>70, "C",C3>60, "D")

Data				
Name	Region	Orders	Sales	IFS
Mark	East	75	60000	C
Brian	East	75	60000	C
Alan	South	86	34567	B
Tony	North	89	24567	B
Agatha	East	68	52908	D
Lana	North	65	13400	D
Heather	East	78	78	C
Ben	East	45	34567	Need_Improvment
Caitlyn	west	78	68907	C
Gibbs	East	90	94000	B

UNIQUE:

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

Syntax: =UNIQUE(array,[by_col],[exactly_once])

- **Array** (required) - the range or array from which to return unique values.
- **By_col** (optional) - a logical value indicating how to compare data:
  - TRUE - compares data across columns.
  - FALSE or omitted (default) - compares data across rows.
- **Exactly_once** (optional) - a logical value that defines what values are considered unique:
  - TRUE - returns values that occur only once, which is the database notion of unique.
  - FALSE or omitted (default) - returns all distinct (different) values in the range or array.

4. **Use Unique function to create the List Unique regions**

```
=UNIQUE(B3:B12)
=unique(B3:D4,false)  /* Compress data across rows*/
=unique(C9:D9,True)  /* Compress data across Columns*
```

Data							
Name	Region	Orders	Sales	UNIQUE	USING UNIQUE FUNCTION BY USING ROWS		
Mark	East	75	60000	East	East	75	60000
Brian	East	75	60000	South			
Alan	South	86	34567	North	USING UNIQUE FUNCTION BY USING columns		
Tony	North	89	24567	west	78		
Agatha	East	68	52908				
Lana	North	65	13400				
Heather	East	78	78				
Ben	East	45	34567				
Caitlyn	west	78	68907				
Gibbs	East	90	94000				

COUNTIFS:

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

Syntax =COUNTIFS(**criteria_range1**, **criteria1**, **[criteria_range2, criteria2]**, ...)

- The criteria_range1, criteria_range2, and so on, are the ranges where the function check for the conditions.
- If a number is **greater than** another number >
- If a number is **smaller than** another number <
- If a number or text is **equal** to something =

SUMIFS Function :

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

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

- The criteria_range1, criteria_range2, and so on, are the ranges where the function check for the conditions.
- 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.

AVERAGEIFS Function:

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

It is typed =AVERAGEIFS:

=AVERAGEIFS (**average_range**, **criteria_range1**, **criteria1, ...**)

The criteria_range1, criteria_range2, and so on, are the ranges where the function check for the conditions.

- 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 average_range is the range where the function calculates the average.

5. **Use COUNTIFS, SUMIFS, AverageIFS, functions to calculate statistics based on multiple Criteria, such as region, orders (>=50), sales.**
   - =COUNTIFS(B$3:B$10;E3;C$3:C$10;F3)
   - =SUMIFS(D$3:D$10;B$3:B$10;E3;C$3:C$10;F3
   - =AVERAGEIFS(D$3:D$23;B$3:B$23;E3;C$3:C$23;F3)

Data				Count_IFS		
Name	Region	Orders	Sales	Regions	Orders	People
Mark	East	75	65000	East	>=50	3
Brian	West	65	49987	West	>=50	1
Alan	South	56	34567	South	>=50	1
Tony	North	78	24567	North	>=50	1
Agatha	East	68	52908			
Lana	North	12	13400			
Heather	East	34	26786			
Ben	East	56	34567			

Data				z		SUM_IFS
Name	Region	Orders	Sales	Regions	Orders	Total Sales for matching Orders
Mark	East	75	65000	East	>=50	152475
Brian	West	65	49987	West	>=50	49987
Alan	South	56	34567	South	>=50	34567
Tony	North	78	24567	North	>=50	24567
Agatha	East	68	52908			
Lana	North	12	13400			
Heather	East	34	26786			
Ben	East	56	34567			

Data						
Name	Region	Orders	Sales	Regions	Orders	Average_IFS
Mark	East	75	65000	East	>=50	50825.0
Brian	West	65	49987	West	>=50	49987.0
Alan	South	56	34567	South	>=50	34567.0
Tony	North	67	24567	North	>=50	24567.0
Agatha	East	68	52908			
Lana	North	12	13400			
Heather	East	34	26786			
Ben	East	56	34567			

LAB PROGRAM 3

VLOOKUP, HLOOKUP, XLOOKUP, COUNT, COUNTA

VLookup In Excel :

- This stands for the vertical lookup that is responsible for looking for a particular value in the leftmost column of a table. It then returns a value in the same row from a column you specify.
- Syntax: VLOOKUP(lookup_value, table_array, col_index_num,[range_lookup]
- lookup_value - This is the value that you have to look for in the first column of a table.
- table - This indicates the table from which the value is retrieved.
- col_index - The column in the table from the value is to be retrieved. The Count Of first Column is always 1.
- range_lookup - [optional] to find an Exact match enter False To find an appropriate match enter True.
- TRUE = approximate match (default). FALSE = exact match.

1. **Now use VlookUp to find Orders of a Specific Employee Based on their name.**
   - **=VLOOKUP(A2;A1:E8;3)**
   - **=VlookUp ("Agatha";A1:H22;6)**

Name	Joining Date	Department	Salary	Orders	Sales	Vlookup	
Mark	31-Dec-21	HUMAN RESOURCES	50,000	75	65000	HUMAN RESOURCES	
Brian	31-Dec-21	Sales	45,000	65	49987		
Alan	14-Jan-22	Legal	25,000	43	34567	Vlookup	
Tony	14-Jan-22	Retail	25,000	32	24567	Name	ORDERS
Agatha	01-Feb-22	Sales	30,000	68	52908	Agatha	68
Lana	01-Feb-22	Accounting	45,000	12	13400		
Heather	04-Aug-22	Accounting	50,000	34	26786		
Ben	04-Aug-22	Sales	30,000	56	34567		

XLOOKUP :

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])

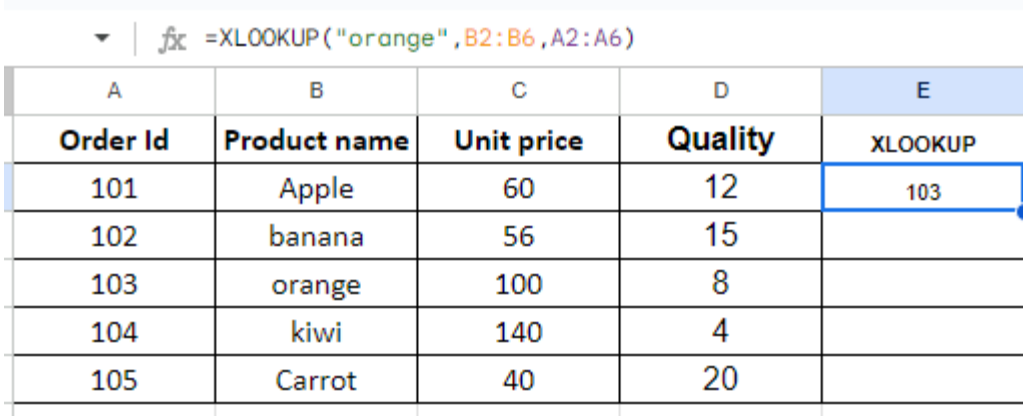
- lookup_value = The value to be searched
- lookup_array =The array or range of cells to search for values
- return_array= The array or range of Sales from which a corresponding Value to be returned, based on the position of value in the look_up array.

if_not_found]	Where a valid match is not found, return the [if_not_found] text you supply.

Optional	If a valid match is not found, and [if_not_found] is missing, **#N/A** is returned.
[match_mode]	Specify the match type:
Optional	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.

2. **Use Xlookup to find the ORDER ID of an particular product based on their product name for vertical table**

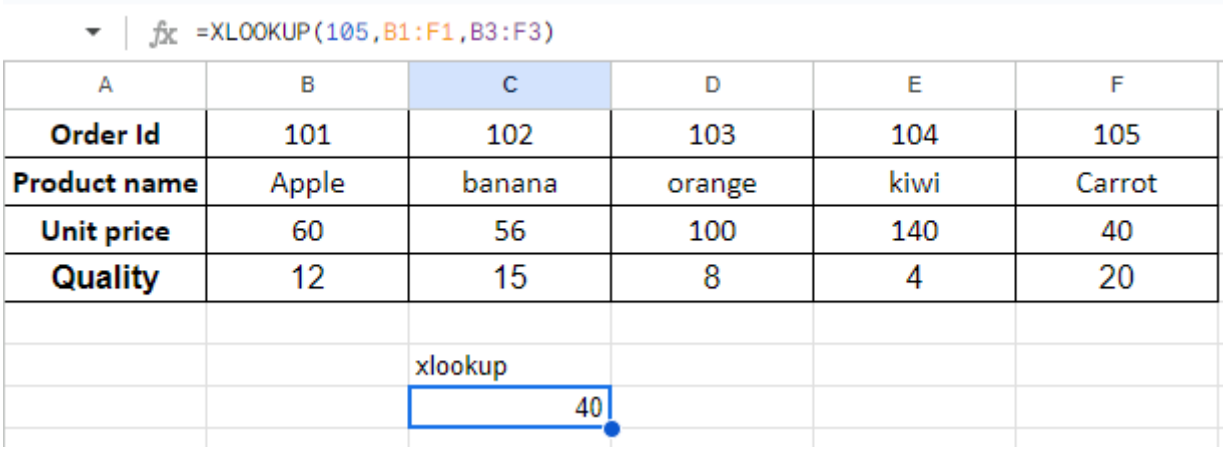
=XLOOKUP (“Orange”, B2:B6, A2:A6)



A	B	C	D	E
Order Id	**Product name**	**Unit price**	**Quality**	XLOOKUP
101	Apple	60	12	103
102	banana	56	15	
103	orange	100	8	
104	kiwi	140	4	
105	Carrot	40	20	

3. **Use Xlookup to find the Unit Price of a product based on their order id in Horizontal Table.**

=XLOOKUP(“105,B1:F1,B3:F3)



A	B	C	D	E	F
Order Id	101	102	103	104	105
Product name	Apple	banana	orange	kiwi	Carrot
Unit price	60	56	100	140	40
Quality	12	15	8	4	20
		xlookup			
		40			

COUNT Explanation:

- The COUNT function is generally used to count a range of cells containing numbers or dates excluding blanks.

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.

- =COUNT(B2:B14)

CountA Function :

- 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 tocount, up to a maximum of 255 arguments.
- 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.

- =COUNTA(B2:B14)

Name	Marks	Count	CountA
Joshi	40	10	13
Nandhan	100		
Radha	78		
Raghu	89		
Ramu	56		
Krishna	34		
Arjun	AB		
Anika	43		
Akshya	98		
Ishwarya	AB		
tejaswini	78		
Pratam	AB		
Hashasri	13/01/23		

Hlookup Function:

- HLOOKUP in Excel stands for 'Horizontal Lookup'. It is a function that makes Excel search for a certain value in a row (the so called 'table array'), in order to return a value from a different row in the same column.

Syntax: HLOOKUP([value], [range], [row number], [false or true])

- The value you want to look up;
- The range in which you want to find the value and the return value;
- The number of the row within your defined range, that contains the return value;
- 0 or FALSE for an exact match with the value your are looking for; 1 or TRUE for an approximate match.

4. **Hlookup Function is used to find out Department of a Specific Employee based on their name.**

Name	Mark	Brian	Alan	Tony	Agatha	Lana	Heather
Joining Date	31-Dec-21	14-Jan-22	01-Feb-22	04-Aug-22	31-Dec-21	14-Jan-22	01-Feb-22
Department	Human Resources	Sales	Legal	Retail	Sales	Accounting	Retail

HlookUp Function

Name	Department
Brian	Sales

Lab Program: 4

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

LEFT Function:

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

Syntax: =LEFT (text,[num_chars])

- **The LEFT function uses the following arguments:**
- **Text** (required argument) – This is the original text string.
- **Num_chars** (optional argument) – Specifies the number of characters to be returned from the start of the given text. It is the number of characters to be extracted, starting from the left side of the text. If we omit the argument, the default value of 1 is taken.
- If you want to use the function on a single cell, write:
  - =LEFT (**cell**)
- If you want to use the function on a range of cells, write:
  - =LEFT(**start cell**:**end cell**)

1. How to Use LEFT Function:

- To retrieve values from the left side of an Excel cell, use **LEFT**

Step 1) Start and Enter the Values by using LEFT function

- Select a cell E2
- Type =LEFT
- Select a cell (A2)
- Hit enter

=LEFT(A2)

2. How to Use the LEFT **Function with a Defined Length of Characters**

1) Select cells and set the number of characters by using LEFT function

1. Select a cell E2
2. Type =LEFT

2) Enter the cell name (A2) and define the length of characters you want to retrieve, using a , as a delimiter (A2,3)

3. Hit enter

=LEFT(A2;3)

- The function returns the first 3 characters from cell A2.

Name	Department	LEFT Function 1	LEFT Function 2	1. The **LEFT** function is used to retrieve a chosen amount of characters, counting from the left side of an Excel cell. The chosen number has to be greater than 0 and is set to 1 by default.
Mark	Human Resources	M	Mar	
Brian	Sales	B	Bri	2. The function returns the first 3 characters from cell A2.
Alan	Legal	A	Ala	
Tony	Retail	T	Ton	
Agatha	Sales	A	Aga	
Lana	Accounting	L	Lan	
Heather	Accounting	H	Hea	
Ben	Sales	B	Ben	
Caitlyn	Retail	C	Cai	

RIGHT Function:

- The function will return a specified number of characters from the end of a given text string.
- In doing financial analysis, the RIGHT function can be useful if we wish to extract characters from the right side of a text string. Generally, it is used by combining it with other functions such as VALUE, SUM, COUNT, DATE, DAY, etc.

Syntax:
=RIGHT (text,[num_chars])

The RIGHT function uses the following arguments:

1. **Text** (required argument) – This is the original text string.
2. **Num_chars** (optional argument) – Specifies the number of characters to be returned from the end of the given text. It is the number of characters to be extracted starting on the right side of the text. If we omit the argument, the default value of 1 is taken.

Example 1:
Suppose we wish to find certain characters from the data below:

- =RIGHT(A2;3)

Example 2 :

- The RIGHT function always returns a text string, even though it contains digits and looks like a number. This is important if we wish to use the result of the function within other functions or formulas.
- Using the VALUE function, we can ensure the number format is kept and the result is not converted to text. we wish to separate numbers from the data below:

- BEN-2345
- JOHN-5432
- BHT-5645

1. We wish to extract the last 4 characters, the order number. The formula to use will be
   - =VALUE(RIGHT(C2;4))

Text	Formula	ORDER Id	Formula
345-111	111	BEN-2345	2345
Abc-5678	5678	JOHN-5432	5432
facebook.com	k.com	BHT-5645	5645

MID Function:

- Generally speaking, the MID function in Excel is designed to pull a substring from the middle of the original text string.
- The Excel MID function has the following arguments:
- = MID(text, start_num, num_chars)

Where:

- **Text** is the original text string.
- **Start_num** is the position of the first character that you want to extract.
- **Num_chars** is the number of characters to extract.

All 3 arguments are required.

1. For example, to pull 3 characters from the text string in A2, starting with the 2nd character, use this formula:

=MID (A2;2;3)

2. MID formula to get the first name and Last name:
   1. =MID(D2;1;SEARCH(" ";D2))
   2. =(MID(D2;SEARCH(" ";D2);LEN(D2)))

	A	B	C	D	E	F
1	**NAME**	**FORMULA**		**NAME**	**FIRST NAME**	**LAST NAME**
2	JOSHI	OSH		K Renuka	K	Renuka
3	SRUTHI	RUT		SAI VISHIKA	SAI	VISHIKA
4	Niraj	ira		SHREE RAM	SHREE	RAM
5	Manoj	ano		JAI KRISHNA	JAI	KRISHNA
6	Kiran	ira		sai roshini	sai	roshini

LEN Function:

LEN returns the number of characters in a text string.

Syntax:

LEN (text)

- **Text** Required. The text whose length you want to find. Spaces count as characters.

=LEN (A2)

	A	B	C	D	E
1	**NAME**	**LEN**		**Subject Name**	**SEARCH**
2	JOSHI	5		DATABASE	5
3	SRUTHI	6		DATA Analytics	1
4	Niraj	5		COMPUTER Network	1
5	Manoj H M	9			
6	Kiran	5			

SEARCH Function:

- 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.
- **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.
- 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")

ISNUMBER Function:

- The function checks if a cell in Excel contains a number or not. It will return TRUE if the value is a number and if not, a FALSE value.
- For example, if the given value is a text, date, or time, it will return FALSE.
    - **Syntax: =ISNUMBER (value)**
- The Excel ISNUMBER function uses the following arguments:
    - **Value** (required argument) – This is the expression or value that needs to be tested. It is generally provided as a cell address.
    - The ISNUMBER Excel function will return a logical value, which is TRUE or FALSE.

Example 1: Let's first understand how the function behaves using the following set of data:

Data	Formula	Result	Remark
1	=ISNUMBER(1)	TRUE	The value provided is a number, so the function returned TRUE.
TEXT	=ISNUMBER(TEXT)	FALSE	The function returns FALSE for text values.
10/20	=ISNUMBER(10/20)	TRUE	The formula will return a number, so the function returned TRUE.
#NAME?	=ISNUMBER(#NAME?)	FALSE	The function returned FALSE for formula errors.
	=ISNUMBER( )	FALSE	The result is FALSE, as it is not a number.

=ISNUMBER (A2)

	A	B
1	**DATA**	**Formula**
2	123	TRUE
3	123-xy	FALSE
4	joshi	FALSE
5		FALSE
6	01/10/20	TRUE
7	01/10/23	TRUE

SUBSTITUTE Function:

- 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.

➢ Copy the example data in the following table, and paste it in cell A1 of a new Excel worksheet.

- Sales Data
- Quarter 1, 2008
- Quarter 1, 2011

FORMULA	DISCRIPTION	RESULT
=SUBSTITUTE(D2, "Sales", "Cost")	Substitutes Cost for Sales (Cost Data)	Cost Data
=SUBSTITUTE(D3, "1", "2", 1)	Substitutes first instance of "1" with "2" (Quarter 2, 2008)	Quarter 2, 2008
=SUBSTITUTE(D4, "1", "2", 3)	Substitutes third instance of "1" with "2" (Quarter 1, 2012)	Quarter 1, 2012

D	E
DATA	FORMULA
Sales Data	Cost Data
Quarter 1, 2008	Quarter 2, 2008
Quarter 1, 2011	Quarter 1, 2012

Lab Program -5
TODAY, NOW, YEAR, MONTH, NETWORKDAYS, EOMONTH

Today Function:

- The TODAY function returns the serial number of the current date in Excel. The TODAY function updates automatically whenever the worksheet is opened, or when the formulas are manually recalculated.

Syntax

- The TODAY function has no arguments.
- The format is: = TODAY()

NOW Function:

- The NOW function in Excel returns the serial number of the current date and time. The NOW function is considered 'volatile', meaning that it updates automatically whenever the worksheet is opened, or when the formulas are manually recalculated.
- **Syntax**
- The NOW function has no arguments.
- The format is: =NOW()
- And ADD days to current date by using =NOW()+7

YEAR Funmction:

- The YEAR function returns the year corresponding to a date. This can be useful for isolating the year element of a date.
- **Syntax**
- The syntax of the YEAR function is: =YEAR(Serial_number)
- **Serial_number** is the Excel-assigned number of the date you want to query. The serial number is usually obtained as the result of another Excel function, such as TODAY, DATE, DATEVALUE, etc.

DATE	FORMULA	RESULT
12-Jan-98	=YEAR(A2)	1998
12/09/23	=year(A3)	2023
20,Jan,2005	=year(DATEVALUE(A4))	2005
03/12/23	=year(today())	2023
2005,6,12	=year(date(2005,6,12))	2005

MONTH Function:

- The MONTH function returns the n^{th} month of the year ranging from 1 to 12. This can be useful for isolating the month element of a date. Or Returns the month of the year a specific date falls in, in numeric format.

Syntax

The syntax of the MONTH function is: =MONTH (Serial_number)

- **Serial_number** is the Excel-assigned number of the date you want to query. The serial number is usually obtained as the result of another Excel function, such as TODAY, DATE, DATEVALUE, etc.

DATE	FORMULA	RESULT
12-Jan-98	=MONTH(A2)	1
12/09/23	=MONTH(A3)	12
20,Jan,2005	=MONTH(DATEVALUE(A4))	1
03/12/23	=MONTH(today())	12
2005,6,12	=MONTH(date(2005,6,12))	6

Note : Month And Year do in google sheet:

- NETWORKDAYS Function :
- NETWORKDAYS is used to determine the number of working days between two dates. The formula calculates the days in an inclusive manner, meaning that the start and end days are included in the count, and Saturdays and Sundays are automatically excluded from the calculation. Additional or ad hoc days may also be excluded using an optional argument.

The syntax of the NETWORKDAYS function is:

- **NETWORKDAYS (start_date, end_date, [holidays])**
- **Start_date** is a date that represents the start date.
- **End_date** is a date that represents the end date.
- **Holidays** is an optional argument. It can refer to a range of dates to be excluded from the working days calculation. Alternatively, **holidays** may be explicitly listed within the formula using an array of date serial numbers.

The basic application of NETWORKDAYS is shown below:

1. =NETWORKDAYS (13-1-2023;31-1-2023;A8:A10) # no of working days b/w 13/01/23 to 31/1/23 and 15/01/23 and 26/01/23 are holidays as a non working days.

2. =NETWORKDAYS(D9;E9) # no of working days b/w 1/12/23 to 31/12/23
3. =NETWORKDAYS(D9;E9;A13) # no of working days b/w 1/12/23 to 31/12/23 and 25/12/23 is holiday as Non working day.

EOMONTH FUNCTION:

- The EOMONTH function determines the last day of an earlier or later month than the month being referenced.
- The EOMONTH function may be used to determine maturity, expiry, or due dates when we want to force these to fall on the last day of the month.
- This function returns the serial number of the calculated date, which can then be formatted using the date format of choice.

Syntax

- EOMONTH(start_date,months)
- Start_date will be used as the reference date.
- Months is the number of months to be calculated before or after the start_date.
  - =EOMONTH(A13;36)
- In the above example, the investment is set to mature on the last date of the month, three years from the investment date.
  - =EOMONTH(A14;0.9)
- If a number other than an integer is entered in **months**, EOMONTH truncates the decimal and uses the whole number value only (see below).
  - =EOMONTH(A18;-1)
- **Months** can also be entered as a negative value, which will result in subtracting that number of months from the start date, as shown below.

	A	B	C
1	**DATE**	**YEAR**	**MONTH**
2	12-Jan-98	1998	1
3	12/09/23	2023	12
4	20,Jan,2005	2005	1
5	03/12/23	2023	12
6	2005,6,12	2005	6

	A	B	C	D	E	F
1	**TODAY**	**NOW**	**NOW**			
2	04/12/23	04/12/23 01:43	08/12/23 01:43	ADD days to current date		
3						
4	**HOLIDAY**	**DATE**	**NETWORKDAYS**			
5	01/01/23	NEW YEAR	14			
6	15/01/23	SANKARANTHI	22	01/01/23	31/01/23	
7	26/01/23	REPUBLIC DAY	21	01/12/23	31/12/23	
8	15/07/23	INDEPENDENCE DAY	20			
9	02/10/23	GANDHI JAYANTHI				
10	25/12/23	CHRISTMAS				
11						
12	**EOMONTH Function**					
13	**Investment date**	**Maturity Date**				
14	16/05/21	31/05/24				
15	16/05/21	31/05/21				
16						
17						
18	**COURSE START DATE**	**APPLICATION DEAD LINE**				
19	12/06/23	31/05/23				
20						

LAB PROGRAM 6: OFFSET, CHOOSE, LET, MAX, SORT, SORTBY, RANK

OFFSET Function: The OFFSET function in Excel returns a cell or range of cells that is a given number of rows and columns from a given cell or range.

The syntax of the OFFSET function is as follows:

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

The first 3 arguments are required and the last 2 are optional.

Required arguments:

Reference - a cell or a range of adjacent cells from which you base the offset. You can think of it as the starting point.

Rows - The number of rows to move from the starting point, up or down. If rows is a positive number, the formula moves below the starting reference, in case of a negative number it goes above the starting reference.

Cols - The number of columns you want the formula to move from the starting point. As well as rows, cols can be positive (to the right of the starting reference) or negative (to the left of the starting reference).

Optional arguments:

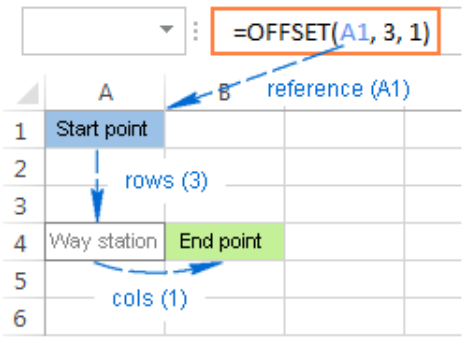
Height - the number of rows to return.

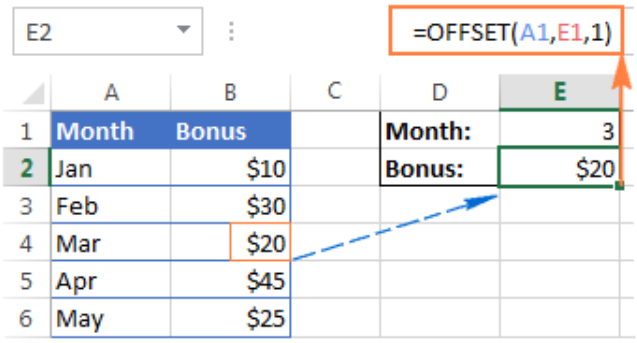
Width - the number of columns to return.

Both the height and width arguments must always be positive numbers. If either is omitted, it defaults to the height or width of *reference*.

OFFSET formula example:

```
=OFFSET(A1,3,1)
```





1. =offset (A1,4,1)
2. =offset (A1,E2,1)
3. =offset (A9,-1,1)
4. =offset (A12,3,1,1,3)
5. =sum(offset(A12,3,1,1,3))

	A	B	C	D	E	F	G	H
1	**Month**	**Bonus**	**Offset function**	**OFFSET FUNCTION**				
2	Jan	10 %	40 %	**Month**	3			
3	Feb	20 %		**Bonus**	30 %			
4	Mar	30 %						
5	Apr	40 %						
6	May	45 %						
7	Jun	50 %		**Month**	**BONUS**			
8	July	55 %		July	55 %			
9	August	60 %						
10								
11								
12	**Name**	**SUB1**	**SUB2**	**SUB3**	**offset**			
13	Arun	100	65	76	78	67	67	
14	Amar	90	45	56	212			
15	priya	78	67	67				
16	Sivaji	65	45	63				
17	Arjun	45	43	73				
18	Prasanth	43	58	86				
19								

Choose function:

The Excel CHOOSE function returns a value from a list using a given position or index.

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.

Choose formula example:

=choose(if(A2=5,1,if(A2=4,2,if(A2=3,3,if(A2=2,4,5)))),"Excellent","GOOD","Average","Poor","Terrible")

	A	B
1	SCORE	Ratings
2	5	Excellent
3	4	GOOD
4	3	Average
5	2	Poor
6	1	Terrible
7		

LET FUNCTION:

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

Syntax

LET has the following syntax and arguments:

LET (name1, name_value1, [name2], [name_value2], …, calculation)

Where:

Name1 (required) - the first name to assign. It must begin with a letter.

Name_value1 (required) - the value or calculation assigned to name1.

Name2 / name_value2 (optional) - the second name and value.

Calculation (required) - a calculation that uses assigned names and values.

The function can process up to 126 name/value pairs.

LET Function Formula:

=let(x,B2,y,B3,x*y) or `=LET(x, 2, y, 5, x*y)`

	A	B
1	LET FUNCTION	
2	x	2
3	y	5
4	PRODUCT	10
5		

MAX Function:

- The MAX function is a premade function in Excel, which finds the highest number in a range. The function ignores cells with text. It will only work for cells with numbers.

Note: There is another function called MIN, which finds the lowest value in a range, the opposite of MAX.

- How to use the =MAX function:
- Type =MAX
- Double click the MAX command
- Select a range (C2:C7)
- Hit enter

Formula : = Max(C2:C7)

	A	B	C	D	E
1	**Name**	**SUB1**	**SUB2**	**SUB3**	**MAX Function**
2	Arun	100	65	76	67
3	Amar	90	45	56	
4	priya	78	67	67	
5	Sivaji	65	45	63	
6	Arjun	45	43	73	
7	Prasanth	43	58	86	
8					

Excel Sorting:

Ranges can be sorted using the Sort Ascending and Sort Descending commands.

Sort Ascending: from smallest to largest.

Sort Descending: from largest to smallest.

The sort commands work for text too, using A-Z order.

Note: To sort a range that has more than one column, the whole range has to be selected.

Sorting just one can breaks the relationship between columns.

Sort in the range **A2:D22** by their **Name**, ascending from smallest to largest (A-Z).

1. Select **A2:D22**
2. Open the **Sort & Filter** menu
3. Click **Sort Ascending**

Note: A1 is not included as it is the header for the column. This is the row that is dedicated to the filter. Including it will blend it with the rest.

=sort (A2:D22)

	A	B	C	D	E	F	G	H
1	**Name**	**Joining Date**	**Department**	**Salary**	**Name**	**Joining Date**	**Department**	**Salary**
2	Mark	31-Dec-21	Human Resources	50,000	Agatha	01-Feb-22	Sales	30,000
3	Brian	31-Dec-21	Sales	45,000	Alan	14-Jan-22	Legal	25,000
4	Alan	14-Jan-22	Legal	25,000	Anderson	01-Mar-22	Sales	24,000
5	Tony	14-Jan-22	Retail	25,000	Ben	04-Aug-22	Sales	30,000
6	Agatha	01-Feb-22	Sales	30,000	Brian	31-Dec-21	Sales	45,000
7	Lana	01-Feb-22	Accounting	45,000	Caitlyn	01-Mar-22	Retail	34,000
8	Heather	04-Aug-22	Accounting	50,000	David	15-Mar-22	Sales	1,25,000
9	Ben	04-Aug-22	Sales	30,000	Gibbs	01-Mar-22	Retail	25,000
10	Caitlyn	01-Mar-22	Retail	34,000	Heather	04-Aug-22	Accounting	50,000
11	Gibbs	01-Mar-22	Retail	25,000	Jack	03-Jan-22	Support	45,000
12	Anderson	01-Mar-22	Sales	24,000	Jacob	15-Mar-22	Support	36,000
13	Michael	01-Mar-22	Retail	1,00,000	Joana	01-Jun-22	Business Development	78,000
14	David	15-Mar-22	Sales	1,25,000	Joey	15-Jul-22	Business Development	56,000
15	Jacob	15-Mar-22	Support	36,000	John	01-Apr-22	Business Development	42,000

Sortby Function :

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

SYNTAX:

SORTBY (array, by_array1, [sort_order1], [by_array2, sort_order2],…)

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

By_array1 (required) - the range or array to sort by.

Sort_order1 (optional) - the sorting order:

1 or omitted (default) - ascending

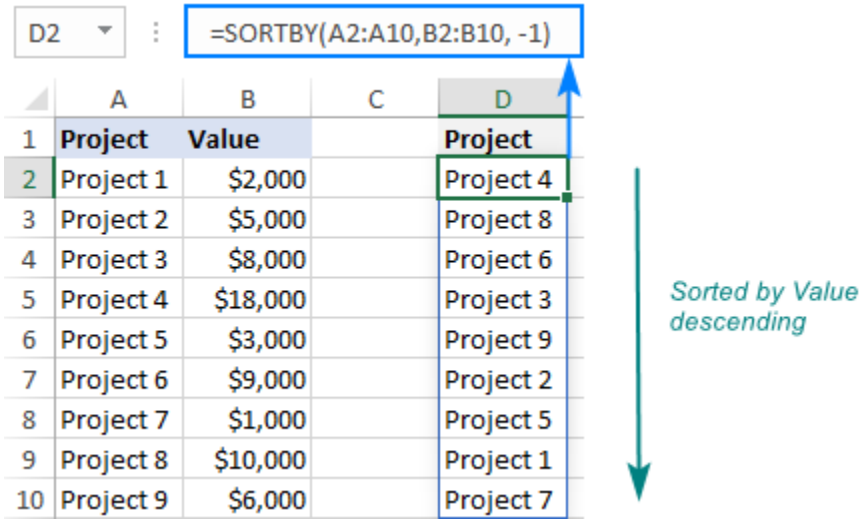
-1 – descending

Important note! Currently the SORTBY function is only available with Microsoft 365 subscriptions and Excel 2021. In Excel 2019, Excel 2016 and earlier versions the SORTBY function is not available.

Formula:

=SORTBY (A2:B10, B2:B10, -1)

- *Array* is A2:A10 - since you do not wish the *Value* column to be displayed in the results, you leave it out of the array.
- *By_array1* is B2:B10 - sort by *Value*.
- *Sort_order1* is -1 - descending, i.e. from highest to lowest.



	A	B	C	D
1	**Project**	**Value**		**Project**
2	Project 1	$2,000		Project 4
3	Project 2	$5,000		Project 8
4	Project 3	$8,000		Project 6
5	Project 4	$18,000		Project 3
6	Project 5	$3,000		Project 9
7	Project 6	$9,000		Project 2
8	Project 7	$1,000		Project 5
9	Project 8	$10,000		Project 1
10	Project 9	$6,000		Project 7

Rank Function:

Returns the rank of a number in a list of numbers. The rank of a number is its size relative to other values in a list. (If you were to sort the list, the rank of the number would be its position.)

Syntax : RANK(number, ref, [order])

The RANK function syntax has the following arguments:

- **Number** Required. The number whose rank you want to find.
- **Ref** Required. An array of, or a reference to, a list of numbers. Nonnumeric values in ref are ignored.
- **Order** Optional. A number specifying how to rank number.

If order is 0 (zero) or omitted, Microsoft Excel ranks number as if ref were a list sorted in descending order. If order is any nonzero value, Microsoft Excel ranks number as if ref were a list sorted in ascending order.

Formula:

=Rank(A2,$A$2:$A$8,1)

=Rank(A2,$A$2:$A$8,0)

	A	B	C	D	E
1	DATA	RANK Function	RANK Function	ASC order	DSC order
2	7	5	3	1	56
3	3.5	4	4	2	34
4	3	3	5	3	7
5	2	2	6	3.5	3.5
6	1	1	7	7	3
7	34	6	2	34	2
8	56	7	1	56	1
9					
10					

Data
7
3.5
3.5
1
2

Formula	**Description (Result)**	**Result**
=RANK(A3,A2:A6,1)	Rank of 3.5 in the list above (3)	3
=RANK(A2,A2:A6,1)	Rank of 7 in the list above (5)	5

Lab Program 7

FILTER, FRQUENCY, SEQUENCE, RANDARRAY, IFERROR

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

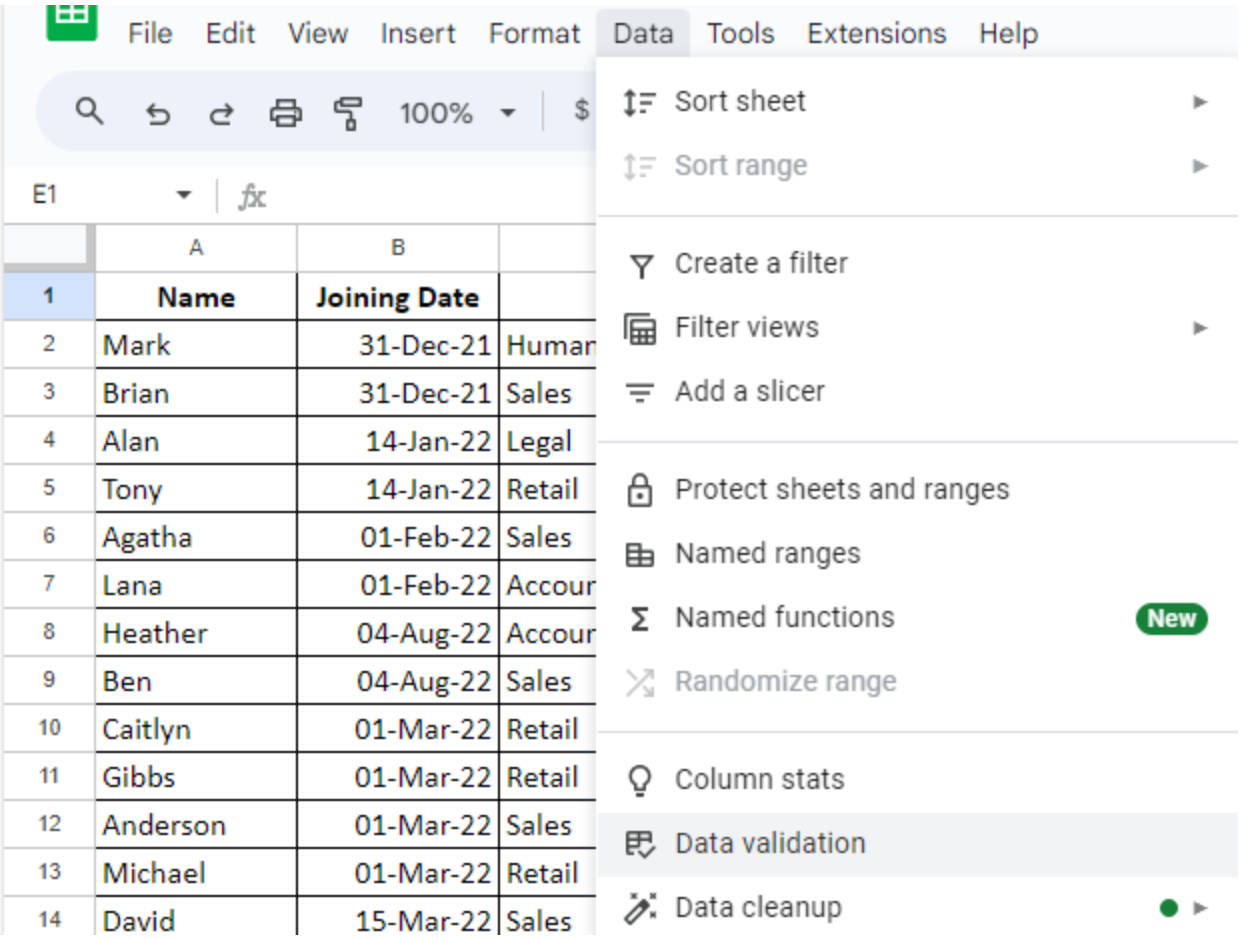
The FILTER function filters an array based on a Boolean (True/False) array.

=FILTER(array,include,[if_empty])

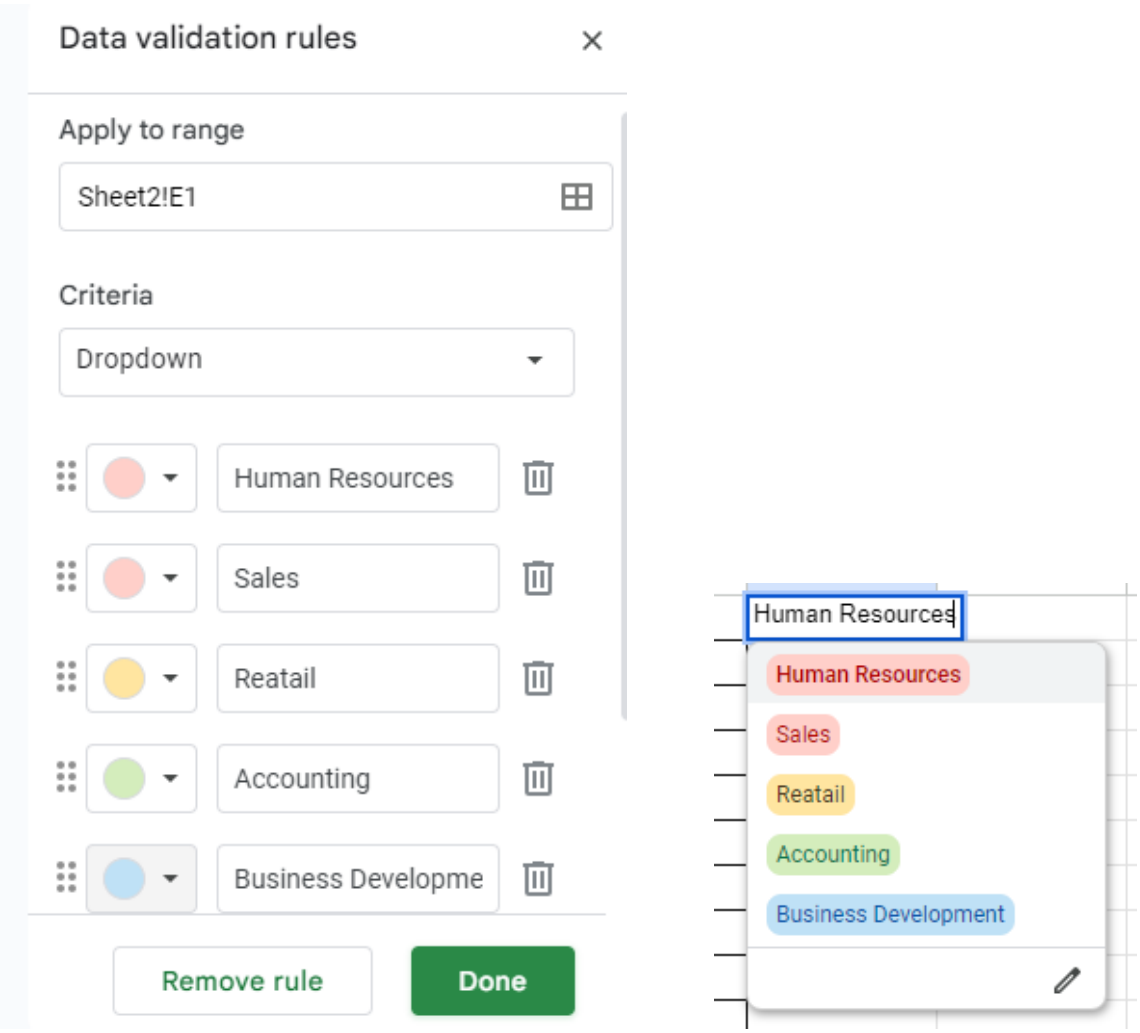
Argument	Description
array Required	The array, or range to filter
include Required	A Boolean array whose height or width is the same as the array
[if_empty] Optional	The value to return if all values in the included array are empty (filter returns nothing)

Step1: In home tab click **data** in data dropdown select **data validation.**

Step2: Now data validation rules window will be open, once it opened add the rule and Click Done.



	A	B	
1	**Name**	**Joining Date**	
2	Mark	31-Dec-21	Humar
3	Brian	31-Dec-21	Sales
4	Alan	14-Jan-22	Legal
5	Tony	14-Jan-22	Retail
6	Agatha	01-Feb-22	Sales
7	Lana	01-Feb-22	Accour
8	Heather	04-Aug-22	Accour
9	Ben	04-Aug-22	Sales
10	Caitlyn	01-Mar-22	Retail
11	Gibbs	01-Mar-22	Retail
12	Anderson	01-Mar-22	Sales
13	Michael	01-Mar-22	Retail
14	David	15-Mar-22	Sales

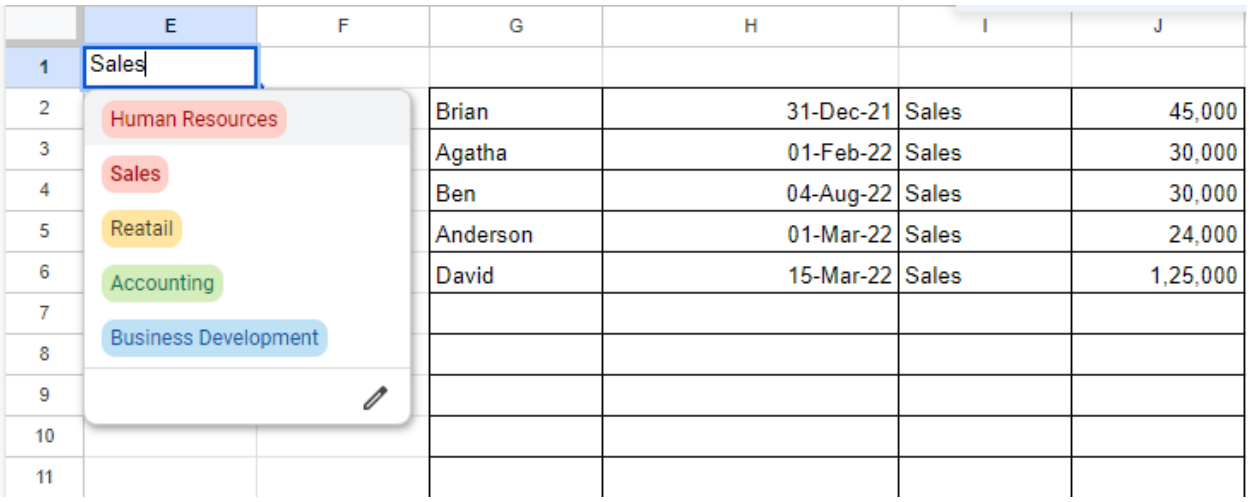


Step 3: now use the filter function to filter the given data by using data validation.

Formula:

=filter (A2:D22,C2:C22=E1)

By using above formula it can filter the data by respective departments.



FREQUENCY function :

The FREQUENCY function calculates how often values occur within a range of values, and then returns a vertical array of numbers. For example, use FREQUENCY to count the number of test scores that fall within ranges of scores. Because FREQUENCY returns an array, it must be entered as an array formula.

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.

	A	B	C
1	**Scores**	**Bins**	
2	79	70	
3	85	79	
4	78	89	
5	85		
6	50		
7	81		
8	95		
9	88		
10	97		
11			
12		**Output**	**Formula Description**
13	**Formula**	**1**	Number of scores less than or equal to 70
14		**2**	Number of scores in the bin 71-79
15		**4**	Number of scores in the bin 80 - 89
16		**2**	number Of scores in the bin 90-97
17			

SEQUENCE Function:

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

In the following example, we created an array that's 4 rows tall by 5 columns wide with **=SEQUENCE (4,5)**.

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

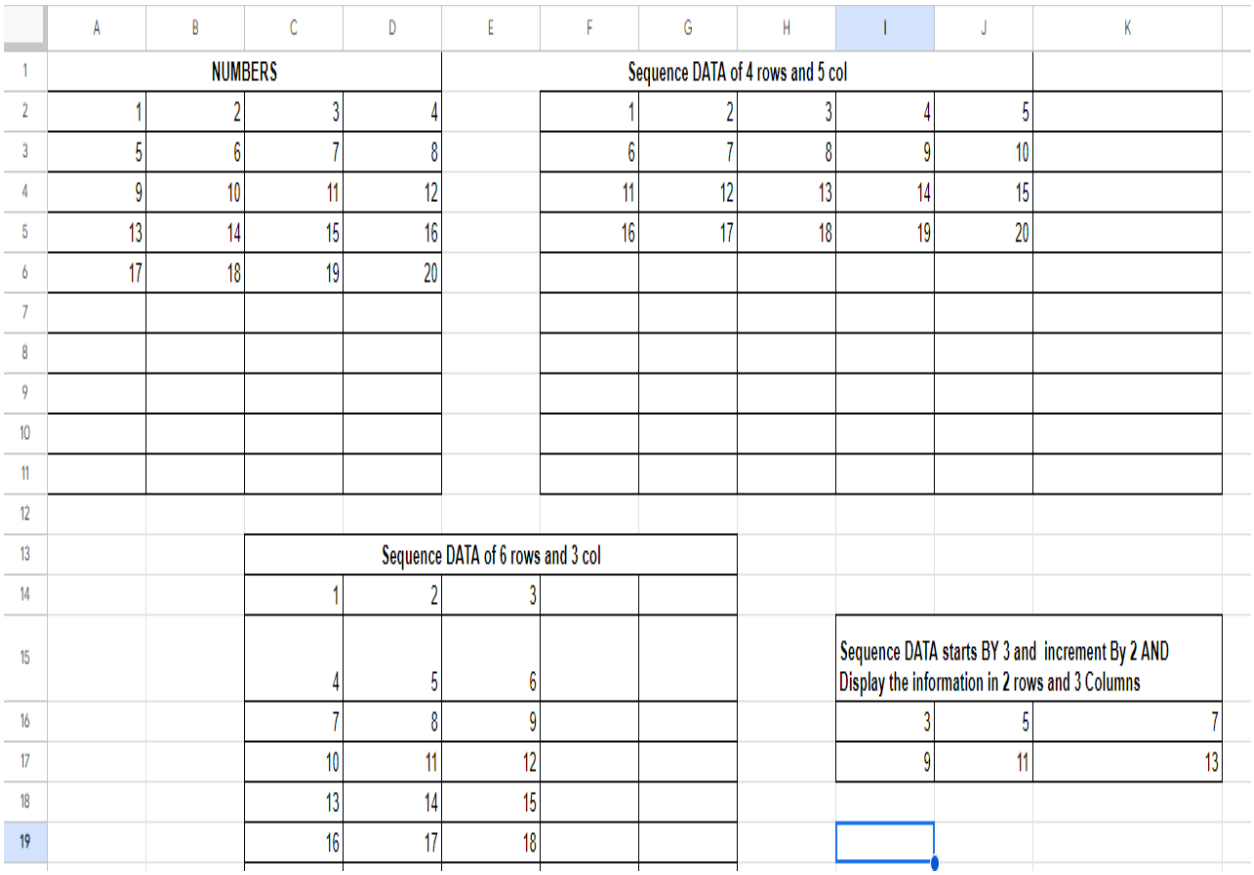
Argument	Description
rows Required	The number of rows to return
[columns]	The number of columns to return

Argument	Description
Optional	
[start] Optional	The first number in the sequence
[step] Optional	The amount to increment each subsequent value in the array

Formula: 1. =SEQUENCE(4,5)

2. =SEQUENCE(6,3)

3. =SEQUENCE(2,3,3,2)



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

Sequence DATA of 4 rows and 5 col				
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

Sequence DATA of 6 rows and 3 col		
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
16	17	18

Sequence DATA starts BY 3 and increment By 2 AND Display the information in 2 rows and 3 Columns		
3	5	7
9	11	13

RANDARRAY function :

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

The RANDARRAY function takes five arguments, none of which are required: *rows*, *columns*, *min*, *max*, and *integer*. By default, *rows*, *columns*, and *max* default to 1, while *min* defaults to zero and *integer* defaults to FALSE. Without any arguments, RANDARRAY will return a decimal value between 0 and 1:

Note: In Google sheet randarray () supports only two arguments. But in excel version 2021 and Microsoft 365 supports 5 arguments.

Formula:

RANDARRAY() // returns number like 0.098419132

2. Use *rows* and *columns* to control the number of values returned:

=RANDARRAY(5,1) // 5 random values in rows

=RANDARRAY(1,5) // 5 random values in columns

	A	B	C	D	E	F	G
1	**Randarray Function 1**				**Randarray Function 2**		
2	0.8949709779				0.09795352147		
3					0.8467596666		
4					0.6599533442		
5	**Randarray Function 4**				0.8482810656		
6	0.1233338939	0.340141546	0.07276152307	0.07179667281	0.1062410858		
7							
8							
9							
10	**Randarray Function 3**						
11	0.6885855937	0.3236578847	0.4960500287	0.2562802149	0.8176867273		
12	0.8541190233	0.5795452901	0.9450752594	0.05376557746	0.8619113542		
13	0.1959916331	0.8435163447	0.3201351375	0.06678692705	0.7074814153		
14	0.7947522538	0.8497343311	0.565997594	0.09701179527	0.67127667		
15	0.5073813413	0.3273603567	0.1752408939	0.068189354	0.5183087512		

IFERROR function:

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!.

Remarks :

- If value or value_if_error is an empty cell, IFERROR treats it as an empty string value ("").
- If value is an array formula, IFERROR returns an array of results for each cell in the range specified in value. See the second example below.

Quota	Units Sold	
210	35	
55	0	
	23	
Formula	**Description**	**Result**
=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	6
=IFERROR(A3/B3, "Error in calculation")	Checks for an error in the formula in the first argument (divide 55 by 0), finds a division by 0 error, and then returns value_if_error	Error in calculation
=IFERROR(A4/B4, "Error in calculation")	Checks for an error in the formula in the first argument (divide "" by 23), finds no error, and then returns the results of the formula.	0

Formula :

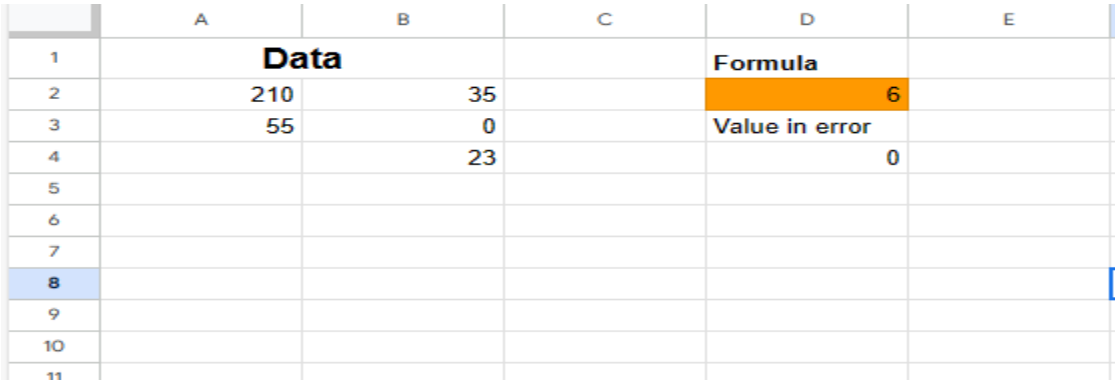
```
=IFERROR(A2/B2)

=IFERROR(A3/B3,"Value in error")

=IFERROR(A4/B4)
```

Example 2

Quota	Units Sold	Ratio
210	35	6
55	0	Error in calculation
	23	0
Formula	**Description**	**Result**
=C2	Checks for an error in the formula in the first argument in the first element of the array (A2/B2 or divide 210 by 35), finds no error, and then returns the result of the formula	6
=C3	Checks for an error in the formula in the first argument in the second element of the array (A3/B3 or divide 55 by 0), finds a division by 0 error, and then returns value_if_error	Error in calculation
=C4	Checks for an error in the formula in the first argument in the third element of the array (A4/B4 or divide "" by 23), finds no error, and then returns the result of the formula Note: If you have a current version of Microsoft 365, then you can input the formula in the top-left-cell of the output range, then press ENTER to confirm the formula as a dynamic array formula. Otherwise, the formula must be entered as a legacy array formula by first selecting the output range, input the formula in the top-left-cell of the output range, then press CTRL+SHIFT+ENTER to confirm it. Excel inserts curly brackets at the beginning and end of the formula for you. For more information on array formulas, see Guidelines and examples of array formulas.	0



Lab Program – 7:

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

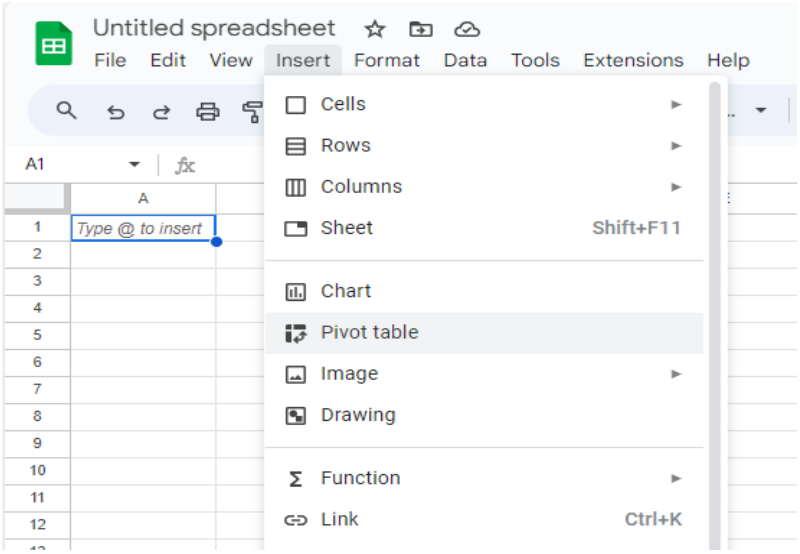
PIVOT TABLES:

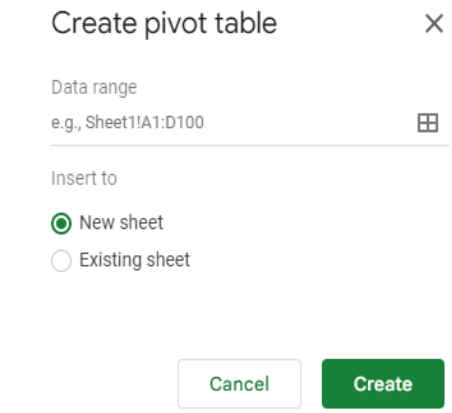
A PivotTable is a powerful tool to calculate, summarize, and analyze data that lets you see comparisons, patterns, and trends in your data. PivotTables work a little bit differently depending on what platform you are using to run Excel.

Step 1: Enter the below data in Google Sheet.

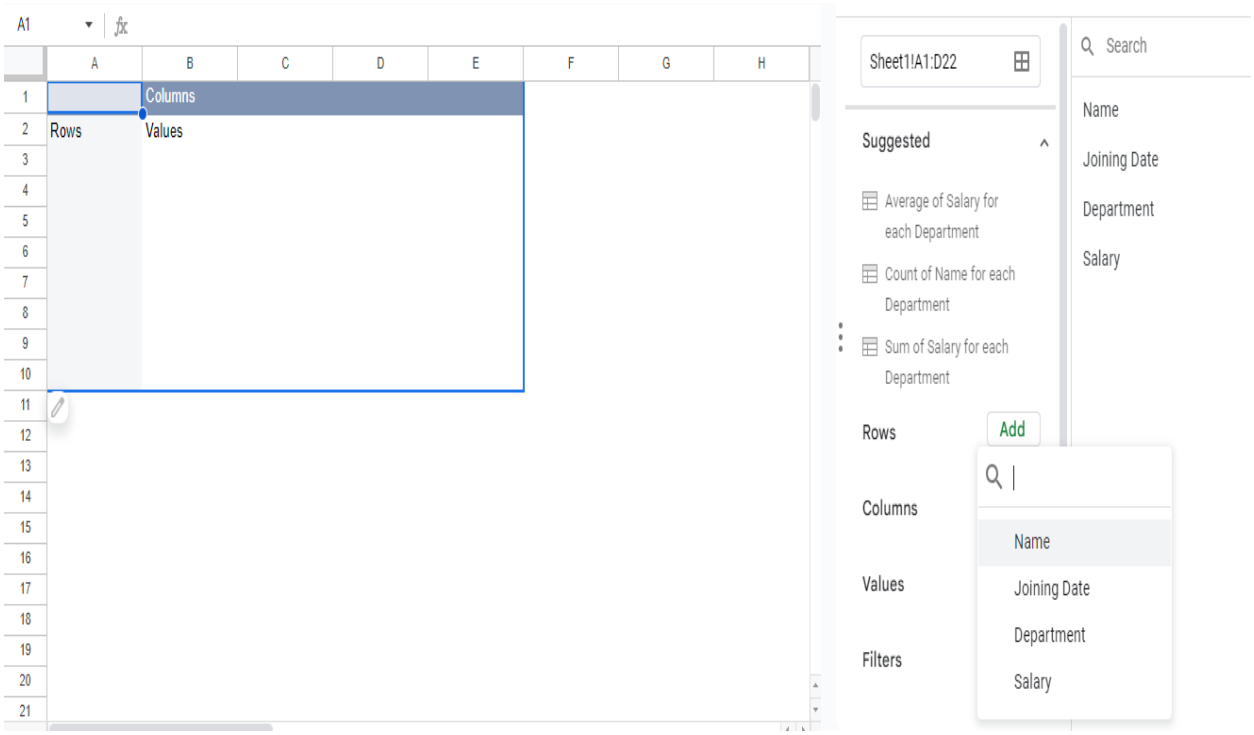
	A	B	C	D
1	**Name**	**Joining Date**	**Department**	**Salary**
2	Mark	31-Dec-21	Human Resources	50,000
3	Brian	31-Dec-21	Sales	45,000
4	Alan	14-Jan-22	Legal	25,000
5	Tony	14-Jan-22	Retail	25,000
6	Agatha	01-Feb-22	Sales	30,000
7	Lana	01-Feb-22	Accounting	45,000
8	Heather	04-Aug-22	Accounting	50,000
9	Ben	04-Aug-22	Sales	30,000
10	Caitlyn	01-Mar-22	Retail	34,000
11	Gibbs	01-Mar-22	Retail	25,000
12	Anderson	01-Mar-22	Sales	24,000
13	Michael	01-Mar-22	Retail	1,00,000
14	David	15-Mar-22	Sales	1,25,000
15	Jacob	15-Mar-22	Support	36,000
16	John	01-Apr-22	Business Development	42,000
17	Leonardo	01-Apr-22	Business Development	65,000
18	Matthew	17-Apr-22	Human Resources	35,000

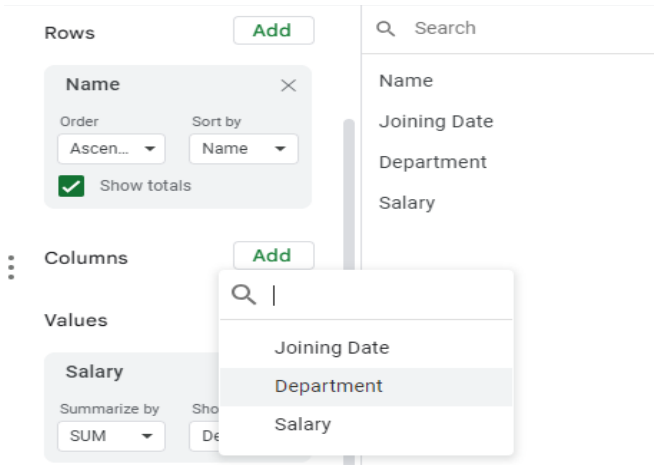
Step 2: Go to insert tab and Click on Pivot Table then it displays the dialogue box.

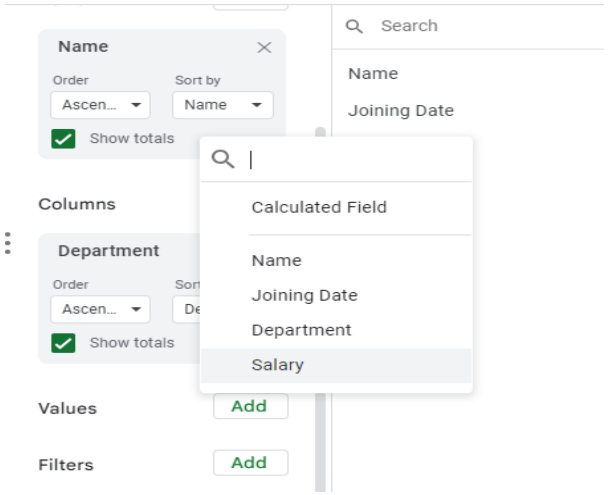




Step 3: select the rows and columns and Values







	A	B	C	D	E	F	G
1	*SUM of Salary*	*Department*					
2	*Name*	Accounting	Business Development	Human Resources	Legal	Retail	Sales
3	Agatha						30,000
4	Alan				25,000		
5	Anderson						24,000
6	Ben						30,000
7	Brian						45,000
8	Caitlyn					34,000	
9	David						0
10	Gibbs					25,000	
11	Heather	50,000					
12	Jack						
13	Jacob						
14	Joana		78,000				
15	Joey		56,000				
16	John		42,000				
17	Lana	45,000					
18	Leonardo		65,000				
19	Mark			50,000			
20	Matthew			35,000			
21	hael					0	

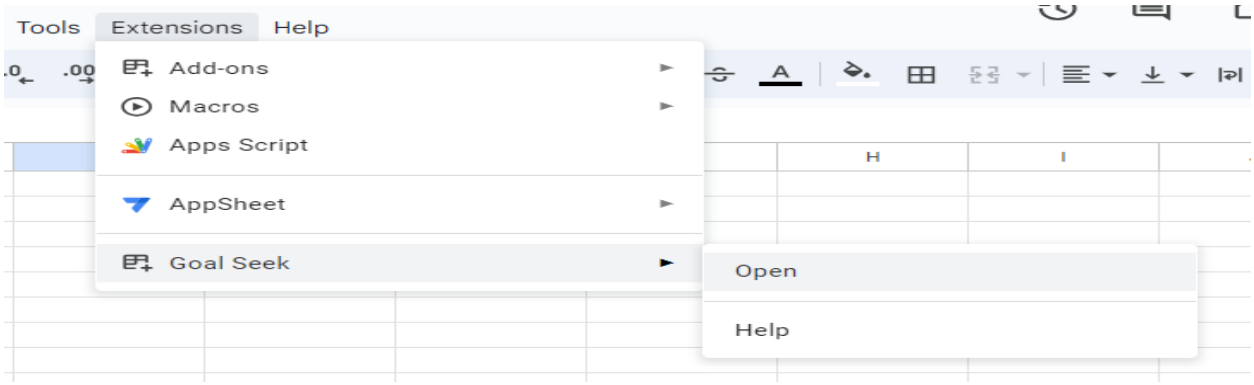
WHAT-IF ANALYSIS :

- 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.

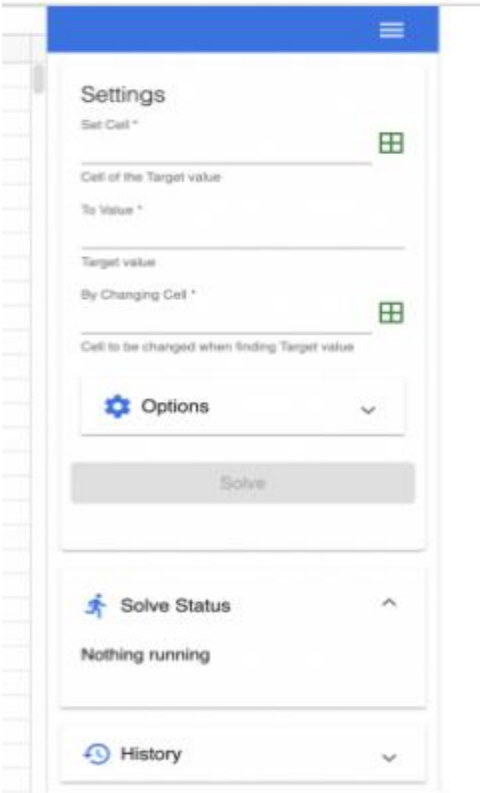
How to Use Goal Seek in Google Sheets :

Install the Goal Seek Add-on in Google Sheets

- Follow the steps below to find and install Goal Seek in Google Sheets.
  - Open Google Sheets and go to Extension > Add-ons > Get add-ons.

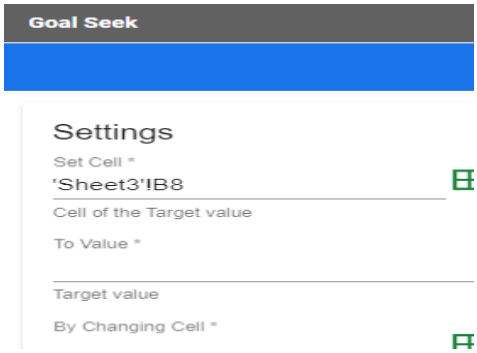


  - You will see the Goal Seek sidebar on the left of your spreadsheet, including "Settings", "Solve Status", and "History".



- Click on the cell containing the profit formula, then click on the grid icon beside "Set Cell" in the sidebar. If you place your cursor over the grid icon, you will see the text "Capture selected cell", as in the screenshot below.

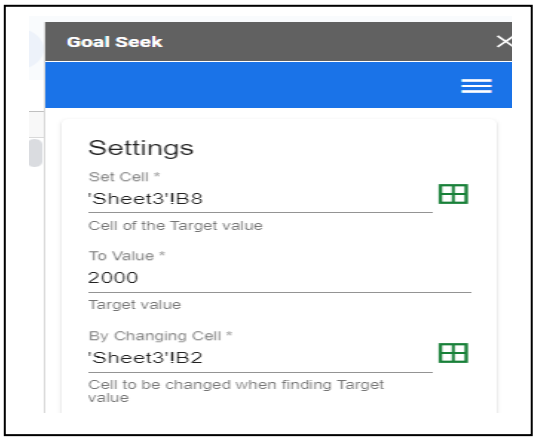
	A	B	C
	What -If Analysis		
	Price	32	
	Qty	100	
	Total Revenue	3200	
	Transport Cost	320	
	Item Cost	2000	
	Total Cost	2320	
	Profit	880	



  - Next, type in the "Target value" for the "Set Cell". Here, "20000".

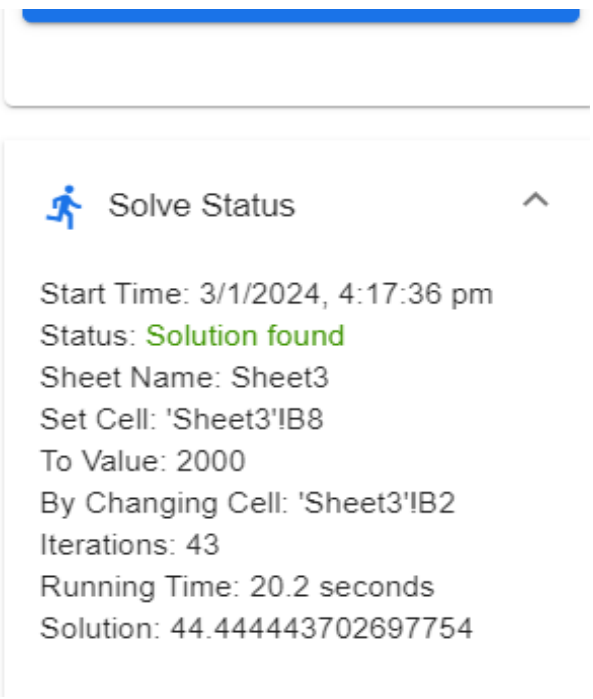
- Click on the cell containing the value you want Goal Seek to change, then click on the grid icon beside "By Changing Cell" in the sidebar. Here, the selling price for Scenario 1, cell B14.

	A	B
1	What -If Analysis	
2	Price	32
3	Qty	100
4	Total Revenue	3200
5	Transport Cost	320
6	Item Cost	2000
7	Total Cost	2320
8	Profit	880
9		



- To change the number of iterations, the tolerance, and/or the time limit, click on "Options". Here, I will use the default values.

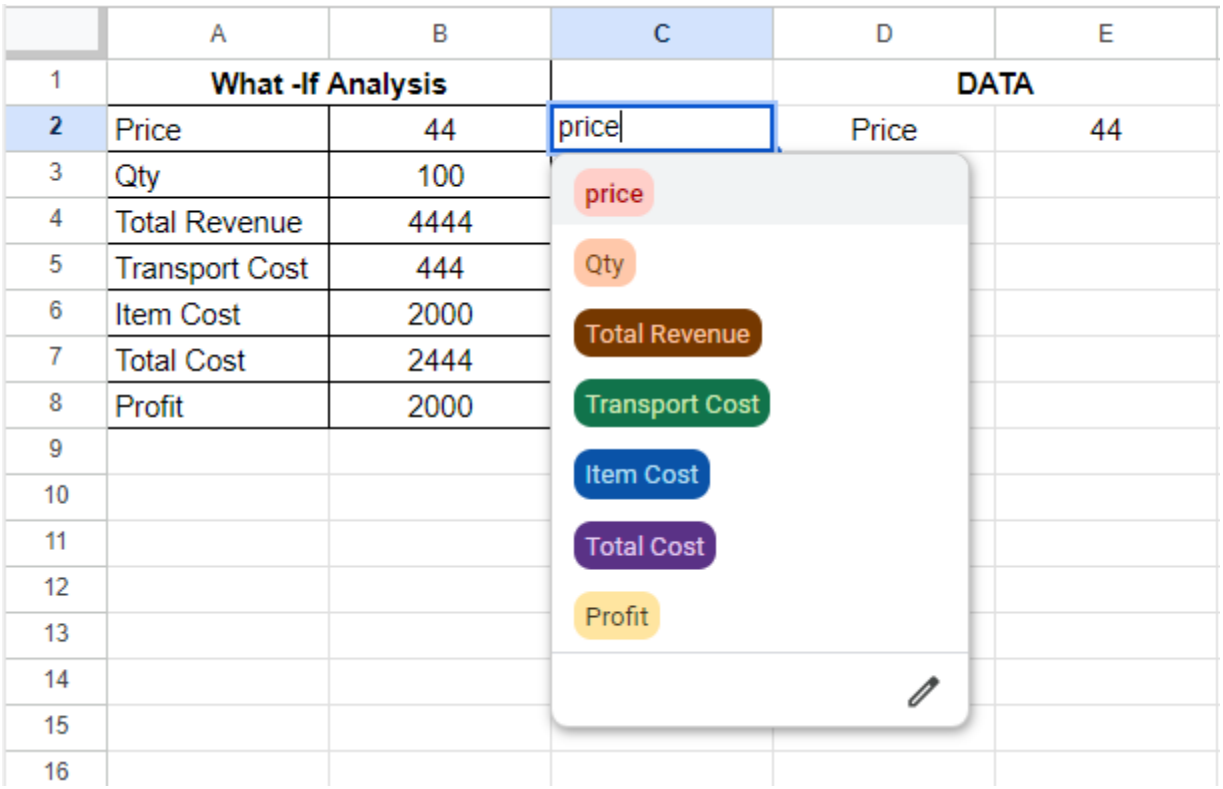
	A	B
1	**What -If Analysis**	
2	Price	44
3	Qty	100
4	Total Revenue	4444
5	Transport Cost	444
6	Item Cost	2000
7	Total Cost	2444
8	Profit	2000
9		



- Once everything is ready, click "Solve." With the default options used above, it will take a maximum of two minutes to calculate the solution. If Goal Seek cannot find a solution, try increasing the tolerance or the time limit and number of iterations.

How to use Data Validation in Google Sheets

- Go to the **"Data"** tab, click **"Data Validation"**, and get a dialog box on the right side of the sheet.
- Click **"+ Add rule"**.
- Select a range where you want to make Data Validation effective.
- Choose one of the criteria and input values according to the criterion.
- Check the advanced option of **"Show help text for a selected cell"**, if you want to leave a hint to an editor who makes an invalid data input to make it valid.
- Choose one of two options* against invalid data input.
- Click **"Done"** to make the setting effective.



SUBTOTALS WITH RANGES

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

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

Part-B: DA Programs:

1. Probability

Program -1:

```
# Simple probability
# Probability of rolling a 4 on a six-sided die
total_outcomes = 6
favorable_outcomes = 1 # Rolling a 4
probability_4 = favorable_outcomes / total_outcomes
print("Probability of rolling a 4:", probability_4)
```

output:

Probability of rolling a 4: 0.16666666666666666

1. A) Calculating the simple probabilities

```
import random
num_trials = int(input("enter_no_of_trials"))
rolls_per_trial =int(input("for Each trail how many rolls"))
roll_up_value=int(input(" Enter rollup value"))
poss_outcomes=0
for i in range(num_trials):
    for j in range(rolls_per_trial):
        result = random.randint(1,6)
        print(result)
        if result == roll_up_value:
```

```
            poss_outcomes += 1
        print("--------")
total_outcomes = num_trials * rolls_per_trial
print(f"Number of times 6 appeared in {num_trials} trials of {rolls_per_trial} rolls each: {poss_outcomes}")
print("probability=",poss_outcomes / total_outcomes)
```

output:

```
enter_no_of_trials 5
for Each trail how many rolls 2
 Enter rollup value 6
3
2
--------
1
2
--------
6
2
--------
6
6
--------
1
5
--------
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**

```
# Binomial Distribution - Decision Making example

# estimating probability of success or failure in fixed number of trials

from scipy.stats import binom

n = 10 # Number of trials

p = 0.5 # Probability of success

k_success = 2# Number of 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

Program 2: Test of significance

```
import pandas as pd

from scipy import stats

titanic_data = pd.read_csv('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: 7.255925461999272e-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

a. Scattered diagram, calculating of correlation coefficient

b. Linear regression: fitting, testing model adequacy and prediction

c. Fitting of logistic regression.

Source Code:

```
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}")
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

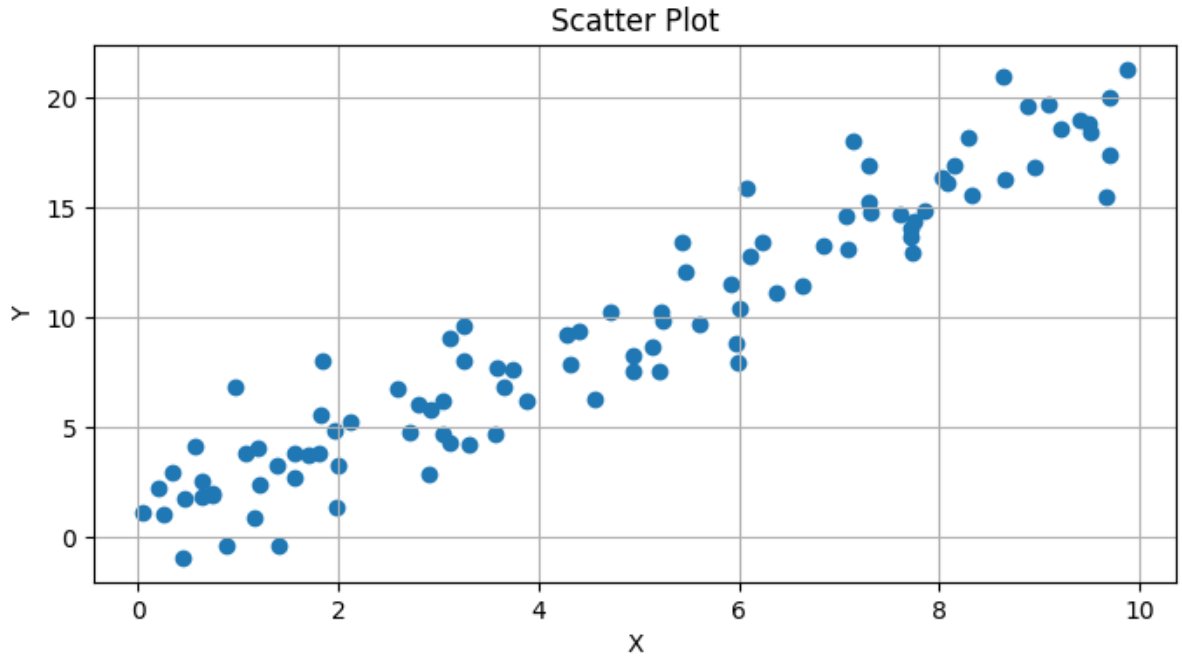
```
x_min, x_max = X_iris[:, 0].min() - 1, X_iris[:, 0].max() + 1
y_min, y_max = X_iris[:, 1].min() - 1, X_iris[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, 0.1), np.arange(y_min, y_max, 0.1))
Z = log_reg.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.figure(figsize=(8, 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)
plt.show()
```

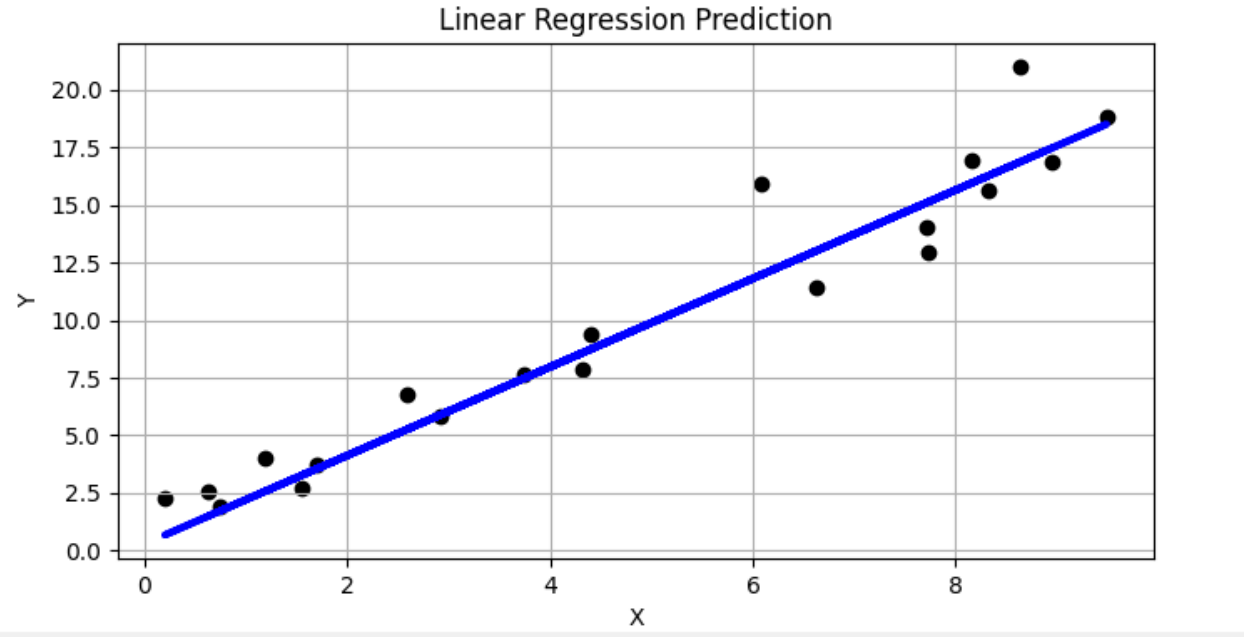
output:

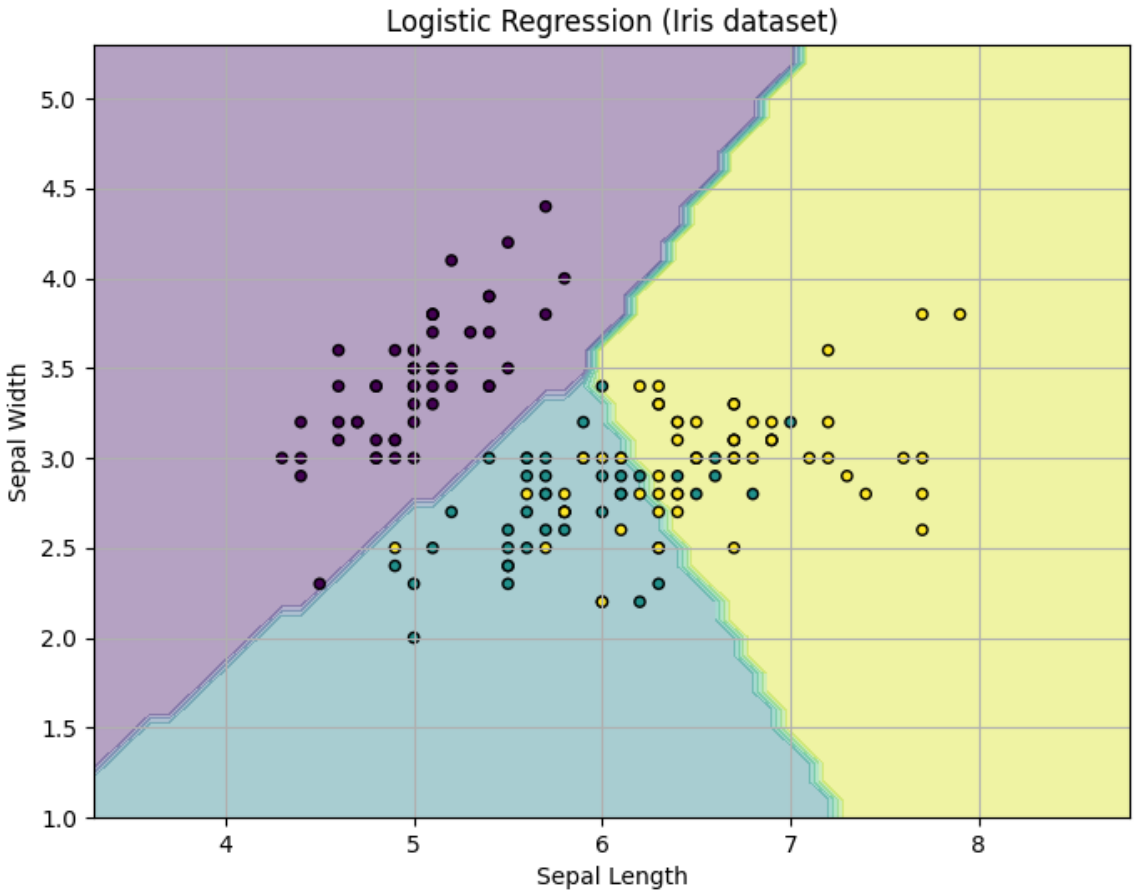
```
Correlation Coefficient: 0.9529657473628446
```

Mean Squared Error: 2.6147980548680088

R-squared Score: 0.9287298556395622






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

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

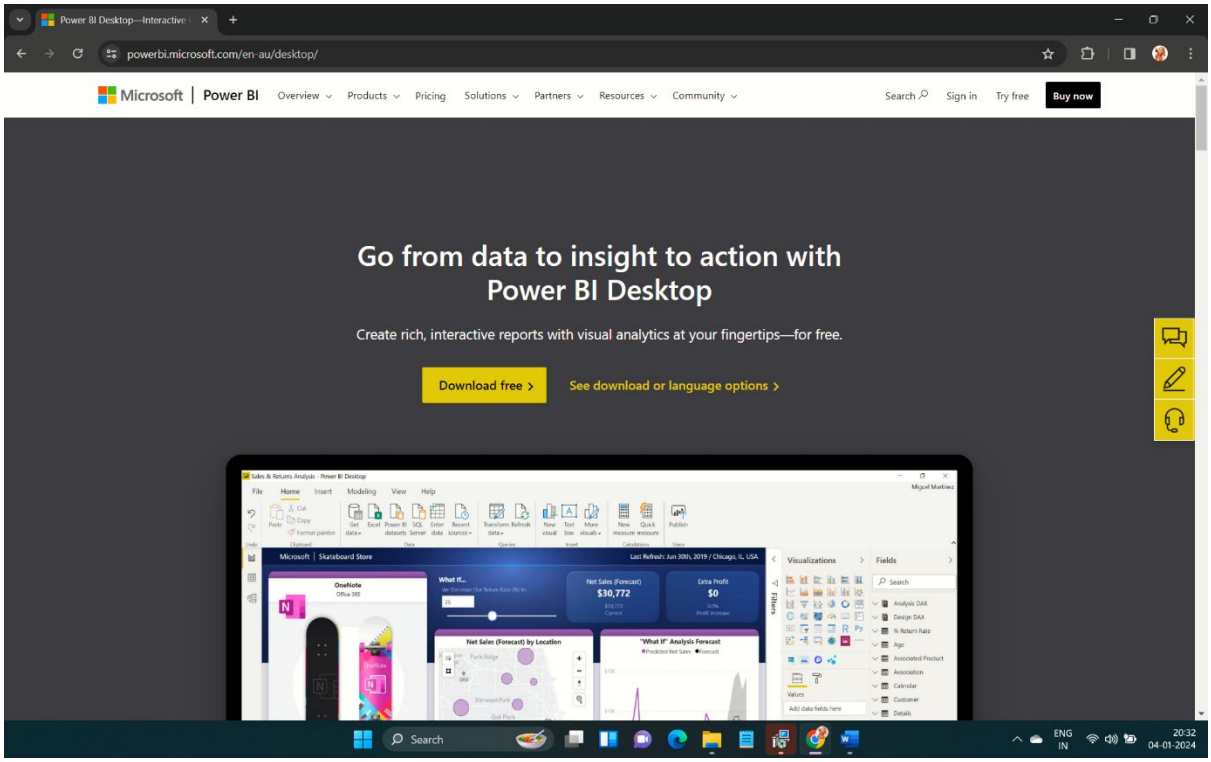
Power BI is a powerful business analytics tool developed by Microsoft that enables users to visualize and analyze data effectively, helping organizations make data-driven decisions.

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.

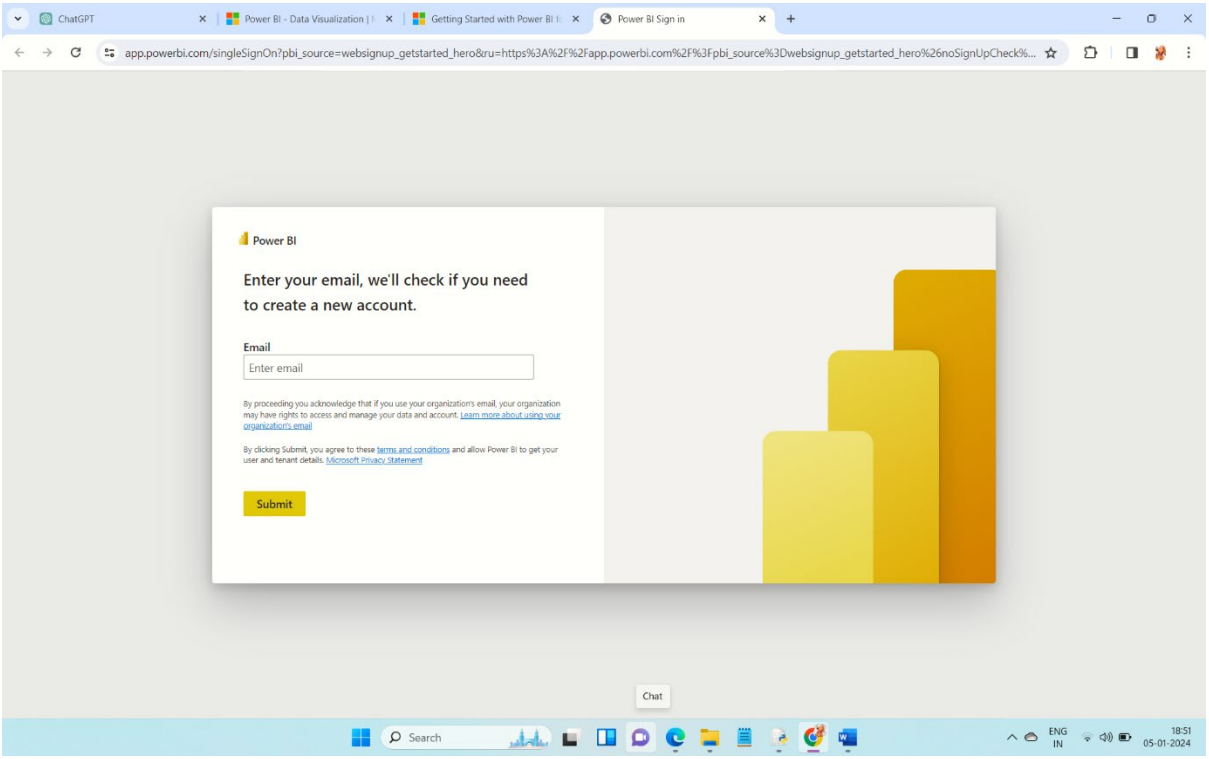
2.Get Started with Power BI

- 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:

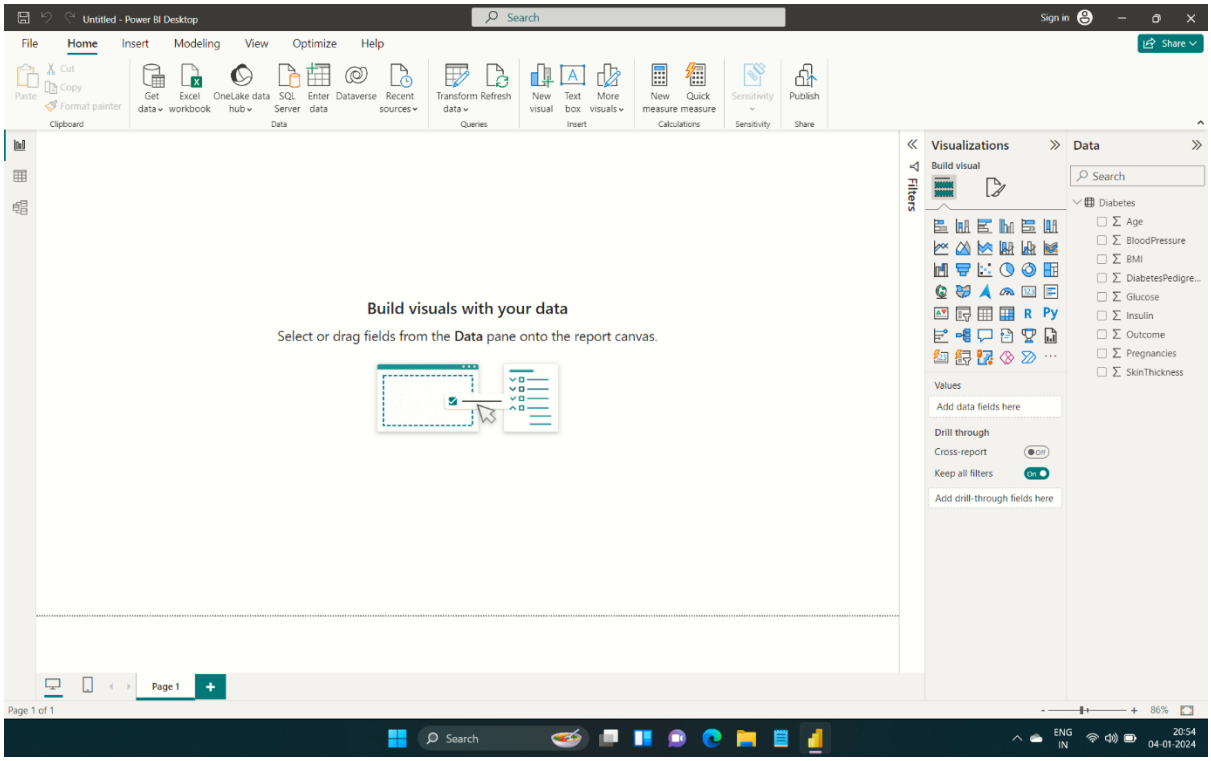
- **SaaS Solutions:** Connect to cloud-based services like Microsoft Dynamics 365, Google Analytics, Salesforce, etc.
- **Local Files:** Upload files from your computer, such as CSV, Excel, JSON, etc.
- **Databases:** Connect to databases like SQL Server, MySQL, PostgreSQL, etc.
- **Web Sources:** Extract 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.

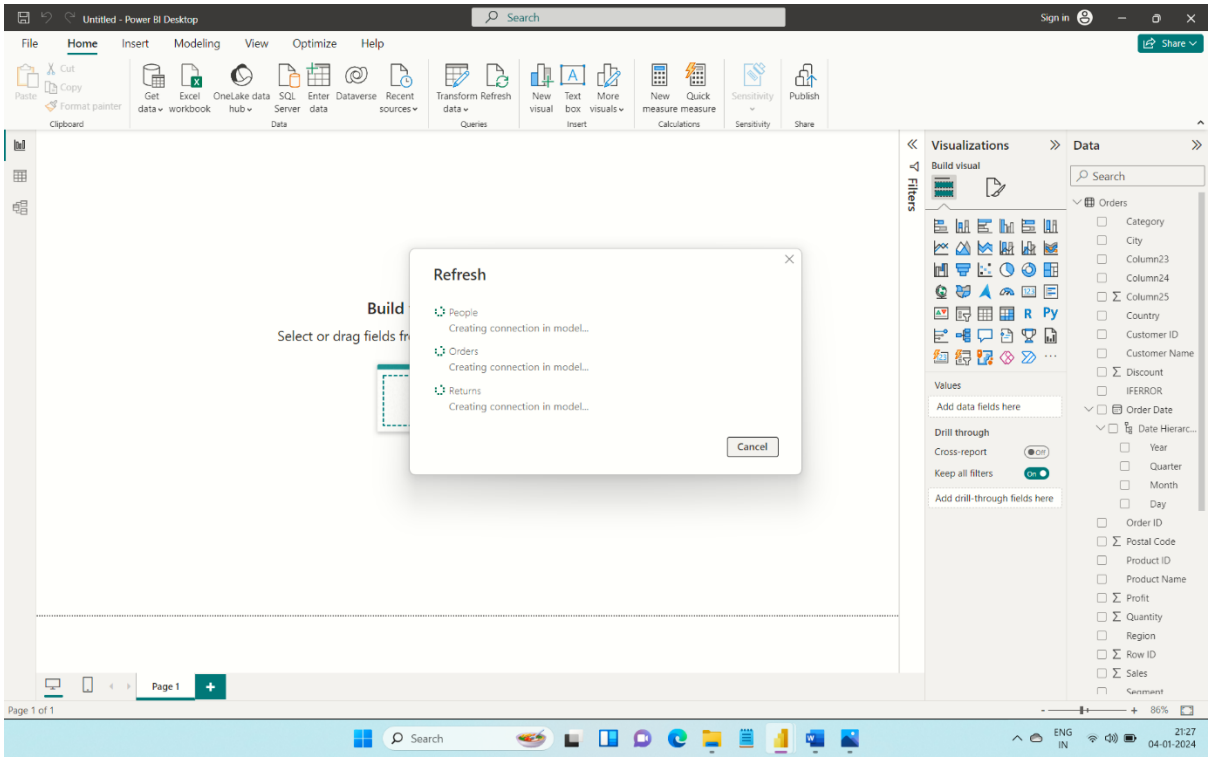
6.Upload a local CSV file:

- 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.



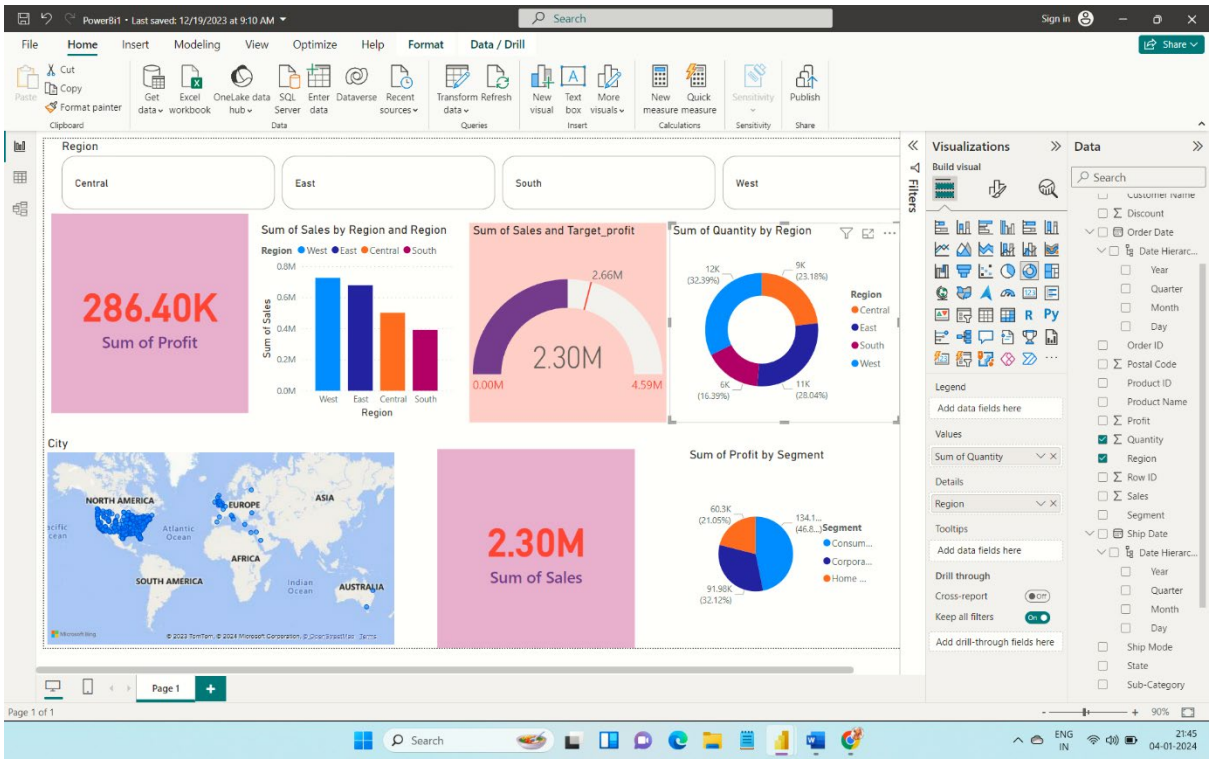
7. Connect to Excel dthat can be refreshed:

- 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:

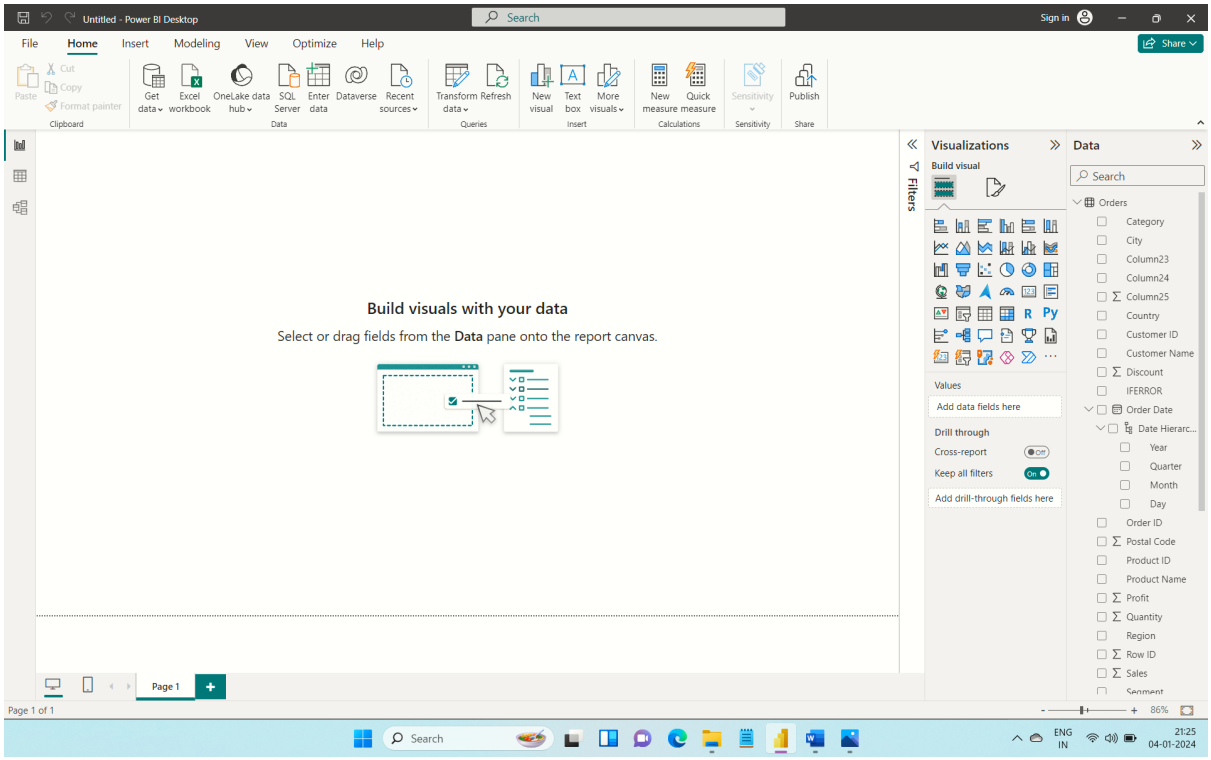
- Select the appropriate type of visualization (e.g., bar chart, line chart, pie chart) from the Visualizations pane on the right-hand side.
- Drag and drop fields from your datasets onto the report canvas to create visualizations such as charts, graphs, tables, etc.
- Customize the visualizations by 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

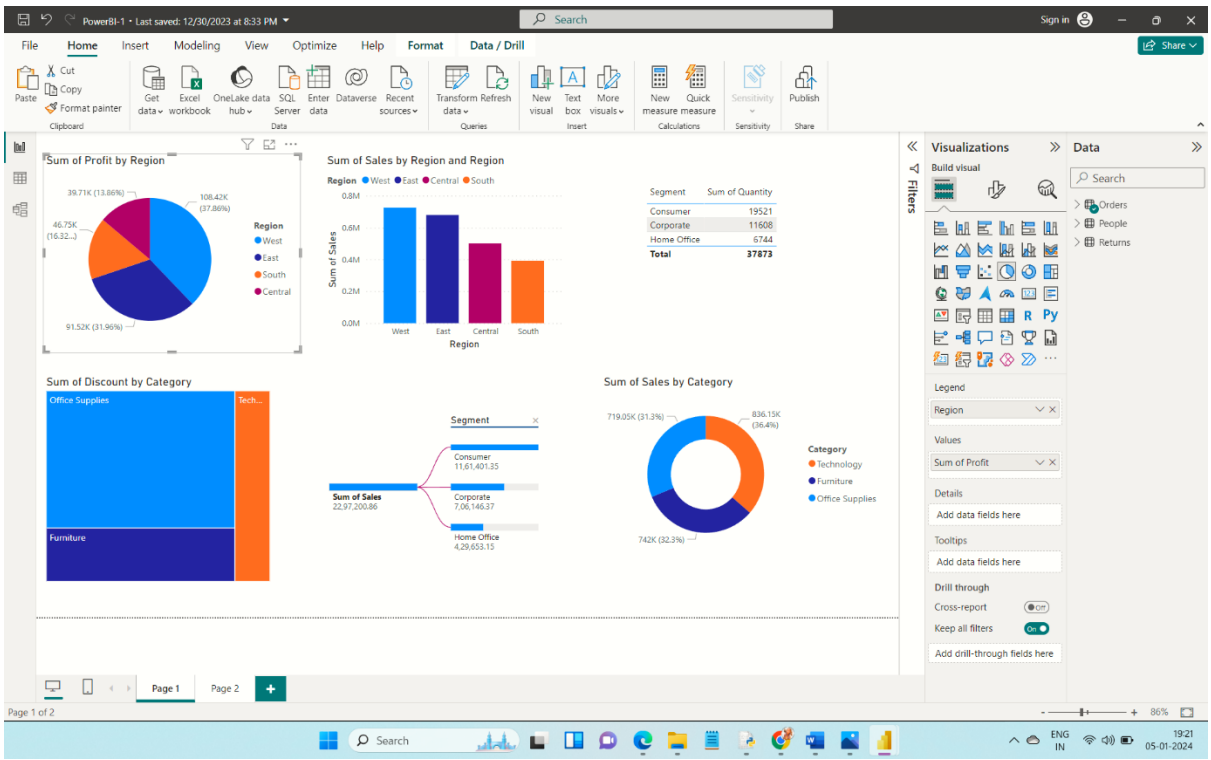
1. Create a New Report:

- Open Microsoft Power BI Desktop.
- Click on "File" > "New"
- Import your data into Power BI by connecting to a data source.



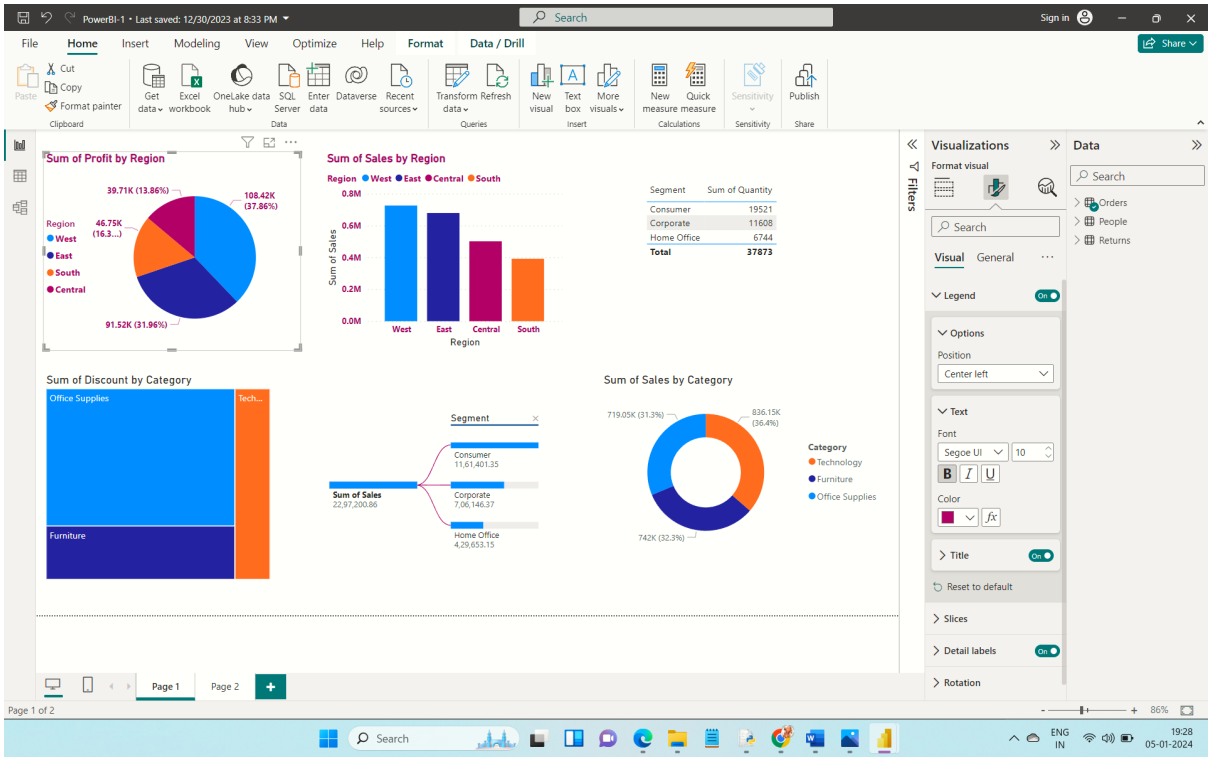
2. Create and Arrange Visualizations:

- 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.



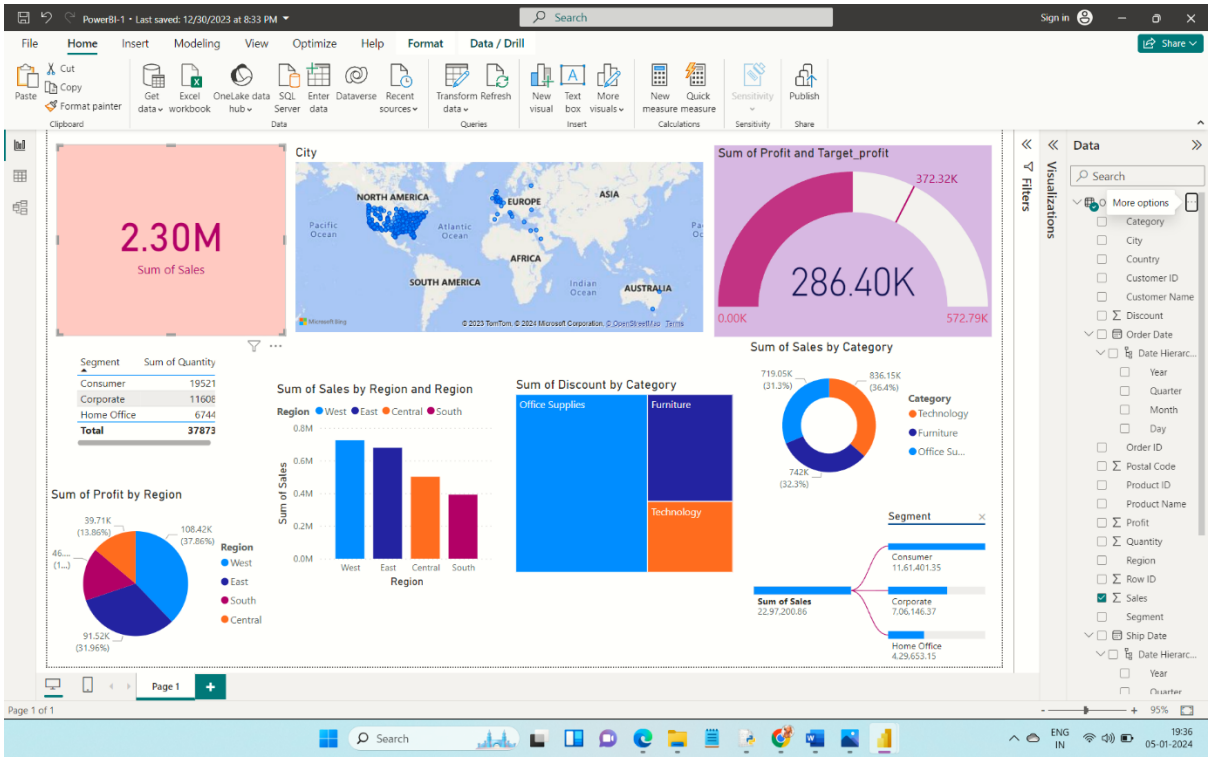
3. Format a Visualization:

- 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.



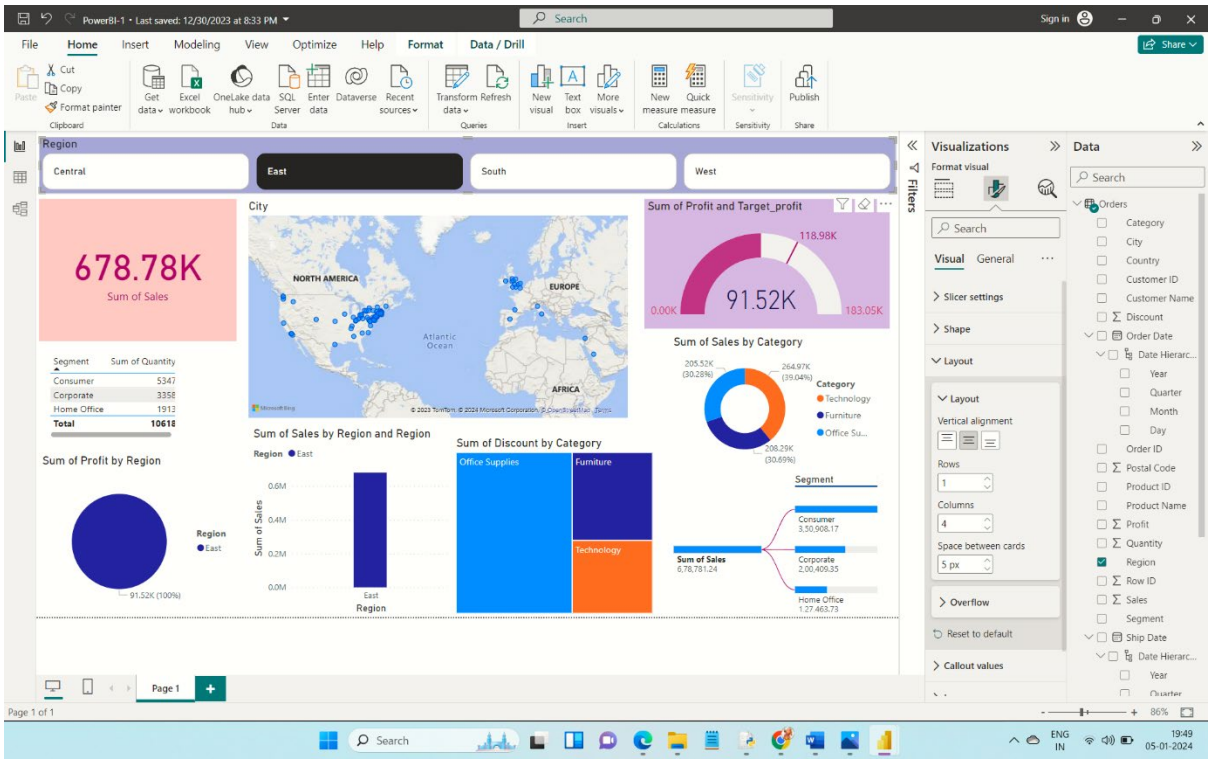
4.Use text, map, and gauge visualizations and save a report:

- 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."

3. 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

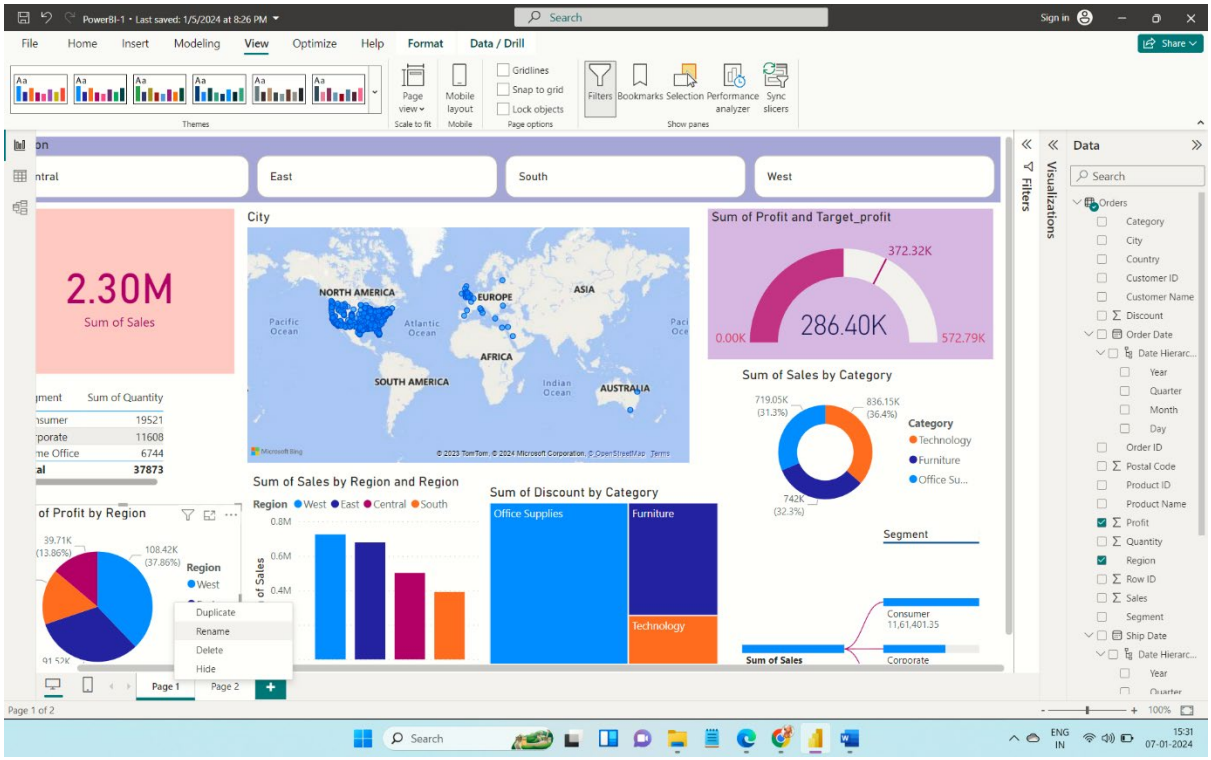
1. Modify and Print a Report:

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.

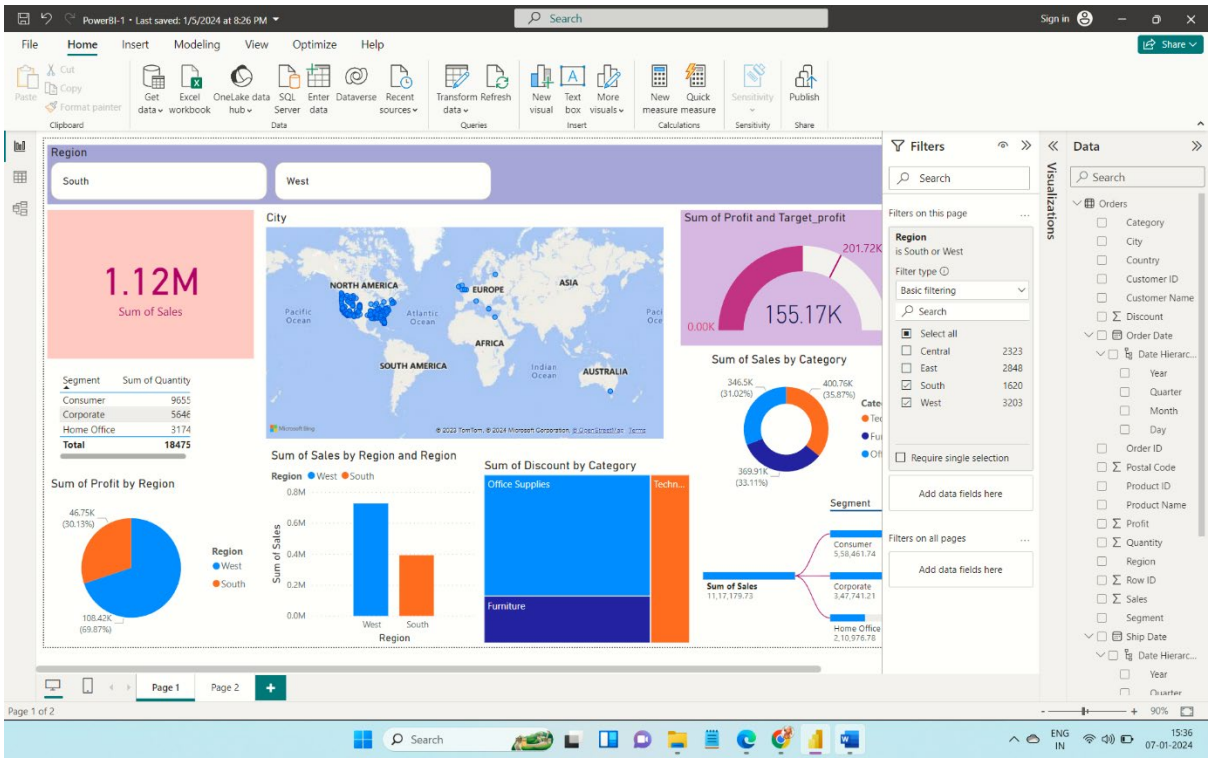
2. Rename and Delete Report Pages:

- 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."



3. 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.

5. 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.