Advance Excel for Data Analysis

Functions and Formulas

REFERENCE

A reference is a way to identify and point to a specific cell or range of cells within a worksheet. It allows you to use the value or content of a cell in formulas, functions, or other calculations.

There are 3 types of references. They are

1. Relative reference
2. Absolute reference
3. Mixed Reference

REFERENCE

1. **Relative Reference:** A relative reference is the default type of reference used in Excel. When you create a formula using relative references, Excel adjusts the reference based on the position of the formula when it is copied or filled to other cells.

   For example, if you have a formula "=A1+B1" in cell C1 and you copy it to cell C2, Excel will automatically adjust the formula to "=A2+B2" because it maintains the relative distance from the original cell.

2. **Absolute Reference:** An absolute reference is used when you want to fix a specific cell or range in a formula, regardless of where the formula is copied or filled. To create an absolute reference, you use the dollar sign ($) before the column letter and/or row number.

   For example, the formula "=$A$1+$B$1" will always refer to cell A1 and B1, even if the formula is copied or filled to other cells.

3. **Mixed Reference** is a combination of both relative and absolute references within a formula. It allows you to fix either the column or row of a cell reference while allowing the other part to change as the formula is copied or filled to other cells.

FORMULA AND FUNCTIONS

Both formulas and functions are used to perform calculations and manipulate data, but there are some key differences between the two

Formulas:

- A formula is an expression that uses mathematical operators, cell references, values, and/or functions to perform calculations.
- You can create your own custom formulas in Excel by combining operators, cell references, and values.
- Formulas begin with an equal sign (=) and can include arithmetic operators (+, -, *, /), comparison operators (=, <, >, etc.), logical operators (AND, OR), and more.
- Formulas are typically entered directly into cells or the formula bar and are visible and editable.
- They allow for more complex and customized calculations and can incorporate multiple functions or perform complex logical operations.
- Examples of formulas: "=A1+B1", "=SUM(A1:A5)", "=IF(A1>10, "Yes", "No")".

Functions:

- A function is a predefined formula or operation that performs a specific calculation.
- Excel provides a wide range of built-in functions that you can use to perform common calculations and manipulate data.
- Functions have predefined syntax and arguments that need to be provided within parentheses.
- Functions can be used to perform various tasks such as arithmetic calculations, statistical analysis, date and time manipulation, text manipulation, and more.
- Examples of functions: "SUM", "AVERAGE", "IF", "VLOOKUP", "CONCATENATE", "DATE", "LEN", etc.
- Functions can be nested within formulas to perform more complex calculations.
- Functions are designed to simplify and automate calculations and are particularly useful for repetitive tasks or complex computations.

SUMIF, COUNTIF, AVERAGE IF

SUMIF: The SUMIF function calculates the sum of a range of cells that meet a given condition or criteria. The syntax of the SUMIF function is as follows:

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

COUNTIF: The COUNTIF function counts the number of cells in a range that meet a specified condition or criteria. The syntax of the COUNTIF function is as follows:

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

AVERAGEIF: The AVERAGEIF function calculates the average of a range of cells that meet a specified condition or criteria. The syntax of the AVERAGEIF function is as follows:

```
=AVERAGEIF(range, criteria, [average_range])
```

Range: The range of cells that you want to evaluate against the criteria.

Criteria: The condition or criteria that the cells must meet. It can be a number, text, logical expression, or cell reference.

Sum_range (optional): The range of cells to be summed if they meet the criteria. If omitted, the function uses the range parameter as the sum_range.

Average_range (optional): The range of cells to be averaged if they meet the criteria. If omitted, the function uses the range parameter as the average_range.

SUMIF and SUMIFS

	SUMIF	SUMIFS
Use	Calculates the sum of a range based on a criteria	Calculates the sum of a range based on multiple criteria
Syntax	SUMIF(range, criteria, [sum_range])	SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
Criteria	Only one criteria can be applied	Multiple criteria can be applied
Criteria Range	Only one range of cells can be evaluated against the criteria	Multiple ranges can be evaluated against the criteria
Sum Range	Optional parameter, if omitted, range will be summed	Required parameter, specifies the range to be summed
Example	SUMIF(A1:A5, ">10") calculates the sum of values greater than 10 in the range A1:A5	SUMIFS(A1:A5, B1:B5, "Apples", A1:A5, ">10") calculates the sum of values greater than 10 for the category "Apples" in the range B1:B5

SUMIFS, COUNTIFS, AVERAGEIFS, MAXIFS, MINIFS

1. **SUMIFS:** The SUMIFS function calculates the sum of a range of cells based on multiple criteria. It allows you to specify conditions or criteria in different ranges to determine which cells to include in the sum. The syntax is as follows:

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

2. **COUNTIFS:** The COUNTIFS function counts the number of cells in a range based on multiple criteria. It allows you to specify conditions or criteria in different ranges to determine which cells to count. The syntax is as follows:

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

3. **AVERAGEIFS:** The AVERAGEIFS function calculates the average of a range of cells based on multiple criteria. It allows you to specify conditions or criteria in different ranges to determine which cells to include in the average. The syntax is as follows:

```
=AVERAGEIFS(average_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
```

4. **MAXIFS:** The MAXIFS function returns the maximum value in a range of cells based on multiple criteria. It allows you to specify conditions or criteria in different ranges to determine which cells to evaluate for the maximum value. The syntax is as follows:

```
=MAXIFS(range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
```

5. **MINIFS:** The MINIFS function returns the minimum value in a range of cells based on multiple criteria. It allows you to specify conditions or criteria in different ranges to determine which cells to evaluate for the minimum value. The syntax is as follows:

```
=MINIFS(range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)
```

VLOOKUP FUNCTION

The VLOOKUP function in Excel is used to search for a specific value in the leftmost column of a table or range. It then retrieves a corresponding value from a specified column within that table. VLOOKUP stands for "Vertical Lookup" because it searches vertically in a column. The syntax of the VLOOKUP function is as follows:

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

lookup_value: This is the value you want to search for in the leftmost column of the table or range.

table_array: This is the range of cells that represents the table or range you want to search. It should include the column containing the lookup value and the column containing the desired result.

col_index_num: This is the column number within the table_array from which you want to retrieve the result. The leftmost column is considered column 1.

range_lookup (optional): This parameter is used to specify whether you want an exact match or an approximate match. If omitted or set to TRUE, an approximate match is used. If set to FALSE or 0, an exact match is required.

VLOOKUP and HLOOKUP

	VLOOKUP	HLOOKUP
Use	Looks up a value vertically in the leftmost column	Looks up a value horizontally in the topmost row
Syntax	VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])	HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])
Lookup Direction	Vertical (downwards)	Horizontal (across)
Lookup Axis	Rows	Columns
Lookup Value	Value to search for in the leftmost column	Value to search for in the topmost row
Table Array	Range of cells that represents the table or range to search	Range of cells that represents the table or range to search
Index Number	Column number within the table array to retrieve the result	Row number within the table array to retrieve the result
Range Lookup	Optional parameter that specifies the match type	Optional parameter that specifies the match type

INDEX and MATCH

	INDEX	MATCH
Use	Retrieves a value from a specific cell in a range	Searches for a value and returns its position in a range
Syntax	INDEX(array, row_num, [column_num])	MATCH(lookup_value, lookup_array, [match_type])
Functionality	Returns the value at the intersection of a specified row and column in a range	Returns the relative position of a value in a range
Search Direction	Can be used for both vertical and horizontal lookups	Typically used for vertical lookups
Lookup Method	Uses the row and column numbers to locate the desired value	Searches for a value and returns its position
Multiple Criteria	Can handle multiple criteria by using arrays and functions	Can be combined with other functions to handle multiple criteria
Exact Match	Requires the exact row and column numbers for retrieval	Allows for both exact and approximate matching
Approximate Match	Can be used for approximate matching with additional functions	N/A (MATCH can only perform exact matches by default)
Flexibility	Provides flexibility to retrieve data from any position in a range	Provides flexibility to search for values in various ways

XLOOKUP FUNCTION

The XLOOKUP function in Excel is a lookup function introduced in Excel 365. It is designed to simplify and enhance the lookup capabilities compared to traditional lookup functions. XLOOKUP is versatile and can perform both vertical and horizontal lookups, handle approximate and exact matches, and handle multiple criteria with ease.

The syntax of the XLOOKUP function is as follows:

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

lookup_value: The value you want to search for.

lookup_array: The range or array where you want to search for the lookup value.

return_array: The range or array from which you want to return a corresponding value.

if_not_found (optional): The value to return if no match is found. By default, it is set to #N/A.

match_mode (optional): Specifies the match mode for approximate matches. It can be 1 (exact match) or -1 (exact match or next smallest value) or 2 (exact match or next largest value). By default, it is set to 1.

search_mode (optional): Specifies the search mode for search behavior. It can be 1 (search from beginning) or -1 (search from end) or 2 (binary search). By default, it is set to 1.

DATE FUNCTIONS

Function	Uses
NETWORKDAYS	Returns the number of whole networkdays (excluding weekends & holidays), between two supplied dates.
NETWORKDAYS.INTL	Returns the number of whole networkdays (excluding weekends & holidays), between two supplied dates, using parameters to specify weekend days (New in Excel 2010).
WORKDAY	Returns a date that is a supplied number of working days (excluding weekends & holidays) ahead of a given start date.
WORKDAY.INTL	Returns a date that is a supplied number of working days (excluding weekends & holidays) ahead of a given start date, using supplied parameters to specify weekend days (New in Excel 2010).
DATE	Returns a date, from a user-supplied year, month and day.
TIME	Returns a time, from a user-supplied hour, minute and second.
TODAY	Returns today's date.
NOW	Returns the current date & time.
HOUR	Returns the hour part of a user-supplied time.
MINUTE	Returns the minute part of a user-supplied time.
SECOND	Returns the seconds part of a user-supplied time.
DAY	Returns the day (of the month) from a user-supplied date.
MONTH	Returns the month from a user-supplied date.
YEAR	Returns the year from a user-supplied date.

NETWORKDAYS FUNCTION

The NETWORKDAYS function in Excel is used to calculate the number of working days between two specified dates, excluding weekends (Saturdays and Sundays) and optionally considering additional specified holidays as non-working days.

The syntax of the NETWORKDAYS function is as follows:

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

start_date: The starting date from which to calculate working days.

end_date: The ending date up to which to calculate working days.

holidays (optional): A range or array of dates that represent additional non-working days, such as public holidays or company-specific holidays.

NETWORKDAYS.INTL FUNCTION

The NETWORKDAYS.INTL function in Excel is an extended version of the NETWORKDAYS function that allows you to calculate the number of working days between two specified dates, taking into account weekends and user-defined non-working days based on a specified custom set of parameters.

The syntax of the NETWORKDAYS.INTL function is as follows:

```
NETWORKDAYS.INTL(start_date, end_date, [weekend], [holidays])
```

start_date: The starting date from which to calculate working days.

end_date: The ending date up to which to calculate working days.

weekend (optional): A number or string that specifies the days of the week that are considered weekends. The default value is 1 (Saturday and Sunday). You can specify a different weekend pattern using a number or a string of binary digits, where each digit represents a day of the week (Monday to Sunday), with 0 indicating a working day and 1 indicating a weekend day.

holidays (optional): A range or array of dates that represent additional non-working days, such as public holidays or company-specific holidays.

WORKDAY FUNCTION

The WORKDAY function in Excel is used to calculate a date that is a specified number of working days (business days) ahead or behind a given date. It allows you to consider weekends (Saturdays and Sundays) and optionally specified holidays as non-working days.

The syntax of the WORKDAY function is as follows:

```
WORKDAY(start_date, days, [holidays])
```

start_date: The starting date from which to calculate.

days: The number of working days to add or subtract from the start_date. Positive values indicate future dates, and negative values indicate past dates.

holidays (optional): A range or array of dates that represent additional non-working days, such as public holidays or company-specific holidays.

WORKDAY.INTL FUNCTION

The WORKDAY.INTL function is an extended version of the WORKDAY function that allows you to calculate a date that is a specified number of working days (business days) ahead or behind a given date, taking into account weekends and user-defined non-working days based on a specified custom set of parameters.

The syntax of the WORKDAY.INTL function is as follows:

```
WORKDAY.INTL(start_date, days, [weekend], [holidays])
```

start_date: The starting date from which to calculate.

days: The number of working days to add or subtract from the start_date. Positive values indicate future dates, and negative values indicate past dates.

weekend (optional): A number or string that specifies the days of the week that are considered weekends. The default value is 1 (Saturday and Sunday). You can specify a different weekend pattern using a number or a string of binary digits, where each digit represents a day of the week (Monday to Sunday), with 0 indicating a working day and 1 indicating a weekend day.

holidays (optional): A range or array of dates that represent additional non-working days, such as public holidays or company-specific holidays.

LOGICAL FUNCTIONS

IF function: The IF function allows you to perform different actions based on a given condition. It follows the syntax:

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

logical_test is the condition or expression that you want to evaluate.

value_if_true is the result or action to be taken if the condition is met (true).

value_if_false is the result or action to be taken if the condition is not met (false).

AND function:

The AND function allows you to check if multiple conditions are all true. It returns TRUE if all conditions are true; otherwise, it returns FALSE. The syntax is as follows:

```
=AND(condition1, condition2, ...)
```

OR function:

The OR function allows you to check if at least one of multiple conditions is true. It returns TRUE if any of the conditions is true; otherwise, it returns FALSE. The syntax is as follows:

```
=OR(condition1, condition2, ...)
```

condition1, condition2, etc., are the conditions or expressions you want to test.

NESTED IF

Nested IF in Excel refers to the practice of using multiple IF functions within one another to create more complex conditional calculations. With nested IF statements, you can evaluate multiple conditions and specify different actions based on the results. The general syntax of a nested IF statement is as follows:

```
=IF(condition1, value_if_true,
    IF(condition2, value_if_true,
        IF(condition3, value_if_true, value_if_false)
    )
)
```

Each nested IF statement is placed as the value_if_true or value_if_false argument of the preceding IF statement. The last value_if_false argument represents the default action if none of the conditions are met.

IFS function can run multiple tests and return a value corresponding to the first TRUE result. Use the IFS function to evaluate multiple conditions without multiple nested IF statements. IFS allows shorter, easier to read formulas.

IFS FUNCTION

The IFS function in Excel is a versatile logical function that allows you to perform multiple conditional tests and return a result based on the first condition that evaluates to TRUE. It eliminates the need for nested IF statements and provides a more streamlined and readable way to handle complex logical evaluations.

The syntax of the IFS function is as follows:

```
IFS(logical_test1, value_if_true1, [logical_test2, value_if_true2],...)
```

logical_test1, logical_test2, and so on: The logical tests or conditions you want to evaluate.

value_if_true1, value_if_true2, and so on: The values or expressions to return if the corresponding logical test is TRUE.

TYPES OF ERRORS

#DIV/0!: This error occurs when a formula attempts to divide a number by zero. For example, if you enter "=5/0" into a cell, Excel will display "#DIV/0!" to indicate the error.

#VALUE!: This error occurs when a formula contains an invalid data type or references cells that contain incompatible data types. It can also occur when a formula references cells that contain error values. For example, if you use a mathematical operator on a cell that contains text instead of a number, Excel will display "#VALUE!".

#REF!: This error occurs when a formula contains a reference to a cell that no longer exists or has been deleted. It can also occur when a formula references a range of cells, but the range is deleted or modified. For example, if you have a formula that refers to cell A1, and you delete column A, Excel will display "#REF!".

#NAME?: This error occurs when Excel does not recognize a text string as a valid function name or cell reference. It can happen if you misspell a function name or if you refer to a cell or range that does not exist. For example, if you enter "=sum(A1:A10)" but accidentally type "smu" instead of "sum", Excel will display "#NAME?".

#NUM!: This error occurs when a formula contains invalid numeric values or arguments. It can happen when you provide incorrect arguments to a function or use a mathematical operator incorrectly. For example, if you enter "=SQRT(-1)" to calculate the square root of a negative number, Excel will display "#NUM!".

TYPES OF ERRORS

#N/A: This error occurs when a formula cannot find the value it is looking for in a lookup operation. It can happen when you use functions like VLOOKUP or HLOOKUP and the lookup value is not found in the specified range. Excel displays "#N/A" to indicate that the value is not available.

#NULL!: This error occurs when the name of the range, haven't added double quotes for a text string, or referred to a range name that doesn't exist or has been deleted. It can also occur when you've used an intersection operator between two non-intersecting range references.

#SPILL!: This error occurs when the spill range for a spilled array formula isn't blank. When the formula is selected, a dashed border will indicate the intended spill range. You can select the Error floatie, and choose the Select Obstructing Cells option to immediately go the obstructing cell(s).

#CALC!: This error occurs when Excel's calculation engine encounters an unspecified calculation error with an array. To resolve it, try rewriting your formula. If you have a nested formula, you can try using the Evaluate Formula tool to identify where the #CALC! error is occurring in your formula.

#BLOCKED: This errors are returned when a required resource can’t be accessed.

IFERROR FUNCTION

The IFERROR function in Excel is a useful tool for handling formula errors. It allows you to specify a value or action to take when a formula results in an error.

The syntax of the IFERROR function is as follows:

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

value: The formula or expression that you want to evaluate

value_if_error: The value or action you want to be returned if the formula results in an error

Examples of the IFERROR function

Replace an error with a custom message:

```
=IFERROR(formula, "Error: Calculation Failed")
```

This formula will evaluate the formula and display the result if it is not an error. If the formula results in an error, it will display the custom error message "Error: Calculation Failed" instead.

Replace an error with a specific value:

=IFERROR(formula, 0)

In this example, if the formula results in an error, the IFERROR function will return the value 0. Otherwise, it will return the calculated result of the formula.

Perform an alternative calculation if an error occurs:

=IFERROR(formula, alternate_formula)

Here, if the formula results in an error, the IFERROR function will instead calculate and return the result of the alternate_formula. If the formula is successful, its result will be returned.