



Excel

For more eBooks please visit:

[corporatefinanceinstitute.com/resources/ebooks](https://corporatefinanceinstitute.com/resources/ebooks)

## About Corporate Finance Institute®

CFI is a world-leading provider of online financial analyst training programs. CFI's courses, programs, and certifications have been **delivered to tens of thousands of individuals** around the world to help them become world-class financial analysts.

The analyst certification program begins where business school ends to teach you job-based skills for corporate finance, investment banking, corporate development, treasury, financial planning and analysis (FP&A), and accounting.

CFI courses have been designed to make the complex simple by distilling large amounts of information into an easy to follow format. Our training will give you the practical skills, templates, and tools necessary to **advance your career and stand out from the competition.**

Our financial analyst training program is suitable for students of various professional backgrounds and is designed to teach you everything from the bottom up. By moving through the three levels of mastery, you can expect to be performing **industry-leading corporate finance analysis** upon successful completion of the courses.

Copyright © 2018 CFI Education Inc.

All rights reserved. No part of this work may be reproduced or used in any form whatsoever, including photocopying, without prior written permission of the publisher.

This book is intended to provide accurate information with regard to the subject matter covered at the time of publication. However, the author and publisher accept no legal responsibility for errors or omissions in the subject matter contained in this book, or the consequences thereof.

Corporate Finance Institute

[learning@corporatefinanceinstitute.com](mailto:learning@corporatefinanceinstitute.com)

1-800-817-7539

[www.corporatefinanceinstitute.com](http://www.corporatefinanceinstitute.com)

At various points in the manual a number of financial analysis issues are examined. The financial analysis implications for these issues, although relatively standard in treatment, remain an opinion of the authors of this manual. No responsibility is assumed for any action taken or inaction as a result of the financial analysis included in the manual.

# Table of contents

|           |   |
|-----------|---|
| <b>6</b>  | <b>Excel Shortcuts for PC and Mac</b>         |
| 7         | Editing Shortcuts                             |
| 8         | Formatting Shortcuts                          |
| 9         | Navigation Shortcuts                          |
| 10        | File Shortcuts                                |
| 10        | Ribbon Shortcuts                              |
| 10        | Paste Special Shortcuts                       |
| 11        | Clear Shortcuts                               |
| 11        | Selection Shortcuts                           |
| 11        | Data Editing Shortcuts                        |
| 12        | Data editing (inside cell) Shortcuts          |
| 12        | Other Shortcuts                               |
| 13        | Reasons to use Excel Shortcuts                |
| 13        | Free Excel Course                             |
| <br>      |   |
| <b>14</b> | <b>Basic Excel Formulas</b>                   |
| 15        | Basic Terms in Excel                          |
| 16        | Five Time-saving Ways to Insert Data in Excel |
| 21        | Seven Basic Excel Formulas For Your Workflow  |
| <br>      |   |
| <b>28</b> | <b>Advanced Excel Formulas</b>                |
| 29        | INDEX MATCH                                   |
| 30        | IF combined with AND / OR                     |
| 31        | OFFSET combined with SUM or AVERAGE           |
| 32        | CHOOSE  |
| 33        | XNPV and XIRR                                 |
| 34        | SUMIF and COUNTIF                             |
| 35        | PMT and IPMT                                  |
| 36        | LEN and TRIM                                  |
| 37        | CONCATENATE                                   |
| 38        | CELL, LEFT, MID and RIGHT functions           |

|     |  |
|-----|--|
| 39  | <b>Most Useful Excel function For Financial Modeling</b> |
| 40  | <b>Date and Time</b>                                     |
| 40  | DATE   |
| 44  | EOMONTH  |
| 49  | TODAY  |
| 52  | YEAR   |
| 56  | YEARFRAC   |
| 60  | <b>Financial</b>   |
| 60  | DURATION   |
| 64  | NPV  |
| 68  | PMT  |
| 71  | PPMT   |
| 75  | XIRR   |
| 78  | XNPV   |
| 84  | YIELD  |
| 88  | <b>Information</b>                                       |
| 88  | ERROR.TYPE   |
| 90  | ISBLANK  |
| 94  | <b>Logical</b>   |
| 94  | AND  |
| 99  | IF   |
| 106 | IFERROR  |
| 112 | IFS  |
| 118 | OR   |
| 124 | CHOOSE   |
| 128 | HLOOKUP  |
| 136 | INDEX  |
| 141 | MATCH  |
| 145 | OFFSET   |
| 150 | VLOOKUP  |
| 157 | <b>Math</b>  |
| 157 | ABS  |
| 162 | SUMIF  |
| 166 | <b>Statistical</b>                                       |
| 166 | AVERAGE  |
| 171 | CORREL   |
| 176 | COUNT  |
| 184 | COUNTA   |
| 191 | COUNTIF  |
| 198 | RANK   |
| 202 | SMALL  |

PART 01

# Excel Shortcuts for PC and Mac



01

02

03

# List of Excel Shortcuts

To learn more, please check out our online courses



## PC shortcuts & Mac shortcuts

### Editing shortcuts


|  |  |  |
|--|--|---|
| Edit active cell   | <b>F2</b>  | <b>Ctrl U</b>   |
| Cut  | <b>Ctrl X</b>  | <b>⌘ X</b>  |
| Copy   | <b>Ctrl C</b>  | <b>⌘ C</b>  |
| Paste  | <b>Ctrl V</b>  | <b>⌘ V</b>  |
| Paste Special  | <b>Alt E S</b>   | <b>⌘ Shift V</b>  |
| Paste name into formula  | <b>F3</b>  | <b>□</b>  |
| Toggle references  | <b>F4</b>  | <b>⌘ T</b>  |
| Start a new line within the same cell  | <b>Alt Enter</b>   | <b>Ctrl Option Return</b>   |
| Insert or edit cell comment  | <b>Shift F2</b>  | <b>Shift F2</b>   |
| Display shortcut menu (i.e. same as right click)   | <b>Shift F10</b>   | <b>Shift F10</b>  |
| Insert worksheet   | <b>Shift F11</b>   | <b>Shift F11</b>  |
| Define a name for a cell   | <b>Ctrl F3</b>   | <b>Ctrl F3</b>  |
| Fill down (e.g. copy formula down in selected cells)   | <b>Ctrl D</b>  | <b>Ctrl D</b>   |
| Fill right   | <b>Ctrl R</b>  | <b>Ctrl R</b>   |
| Insert argument names and parentheses for a function after typing a function name in a formula | <b>Ctrl Shift A</b>  | <b>Ctrl Shift A</b>   |
| Insert row   | <b>Alt I R</b>   | <b>Ctrl I</b>   |
| Insert column  | <b>Alt I C</b>   | <b>Ctrl I</b>   |

## Formatting shortcuts

|  | Windows      | Mac       |
|--|--------------|-----------|
| Brings up format cells menu  | Ctrl 1       | ⌘ 1       |
| Bold   | Ctrl B       | ⌘ B       |
| Italic   | Ctrl I       | ⌘ I       |
| Undo   | Ctrl Z       | ⌘ Z       |
| Repeat last action   | Ctrl Y       | ⌘ Y       |
| Select all used cells (select entire worksheet if command is repeated) | Ctrl A       | ⌘ A       |
| Number format  | Ctrl Shift ! | ⌘ Shift ! |
| Date format  | Ctrl Shift # | ⌘ Shift # |
| Percent format   | Ctrl Shift % | ⌘ Shift % |
| Increase font size   | Alt h fg     | ⌘ Shift > |
| Decrease font size   | Alt h fk     | ⌘ Shift < |
| Increase decimal   | Alt h        | □         |
| Decrease decimal   | Alt h 9      | □         |
| Increase indent  | Alt h 6      | Ctrl M    |
| Decrease indent  | Alt h 5      | ⌘ Shift m |





## Navigation shortcuts

|   |  |  |
|---|--|---|
| Move from cell to cell  | Arrows   | Arrows  |
| Go to   | F5   | F5  |
| Go to cell A1   | Ctrl Home  | Fn Ctrl Left  |
| Go to beginning of row  | Home   | Fn Left   |
| Select the adjacent cell  | Shift Arrow  | Shift Arrow   |
| Select entire row   | Shift Spacebar   | Shift Spacebar  |
| Select entire column  | Ctrl Spacebar  | Ctrl Spacebar   |
| Select all to the start of the sheet  | Ctrl Shift Home  | Ctrl Shift Home   |
| Select all to the last used cell of the sheet   | Ctrl Shift End   | Ctrl Shift End  |
| Select to the end of the last used cell in row/column                                       | Ctrl Shift Arrow   | Ctrl Shift Arrow  |
| Select the last used cell in row/column   | Ctrl Arrow   | Ctrl Arrow  |
| Move one screen up  | PageUp   | Fn Up   |
| Move one screen down  | PageDown   | Fn Down   |
| Move one screen left  | Alt PageUp   | Fn Option Up  |
| Move one screen right   | Alt PageDown   | Fn Option Down  |
| Move to the next or previous worksheet (Move between tabs if you are in a menu window)      | Ctrl PageUp/Down   | Ctrl PageUp/Down  |
| Move to next workbook (while in spreadsheet)<br>Move to next divider (when in menu options) | Ctrl Tab   | Ctrl Tab  |
| Move to the next cell (Move between items within a menu window)                             | Tab  | Tab   |

## File Shortcuts

|                            |  |            |
|----------------------------|--|---|
| New                        | Ctrl N   |  N         |
| Open                       | Ctrl O   |  O         |
| Save workbook              | Ctrl S   |            |
| Save As                    | F12  |  ⇧ Shift S |
| Print                      | Ctrl P   |  P         |
| Open print preview window  | Ctrl F2  | <input type="checkbox"/>  |
| Go to next workbook        | Ctrl Tab   |  -         |
| Close file                 | Ctrl F4  |  W         |
| Close all open Excel files | Alt F4   |  Q         |

## Ribbon Shortcuts

|                              |  |      |
|------------------------------|--|---|
| Show ribbon accelerator keys | Alt  | <input type="checkbox"/>  |
| Show/hide ribbon             | Ctrl F1  |  ~ R |



## Paste Special Shortcuts

|                        |  |           |
|------------------------|--|--|
| Paste Special formats  | Ctrl Alt V t   | Ctrl  V t |
| Paste Special values   | Ctrl Alt V v   | Ctrl  V v |
| Paste Special formulas | Ctrl Alt V f   | Ctrl  V f |
| Paste Special comments | Ctrl Alt V c   | Ctrl  V c |



## Clear Shortcuts

|                                     |  |  |
|-------------------------------------|--|---|
| Clear cell data                     | Delete   | Delete  |
| Clear cell formats                  | Alt H E F  | <input type="checkbox"/>  |
| Clear cell comments                 | Alt H E M  | <input type="checkbox"/>  |
| Clear all (data, formats, comments) | Alt H E A  | <input type="checkbox"/>  |

## Selection Shortcuts

|                                   |  |  |
|-----------------------------------|--|---|
| Select a cell range               | Shift Arrows   | Shift Arrows  |
| Highlight a contiguous range      | Ctrl Shift Arrows  | Ctrl Shift Arrows   |
| Extend selection up one screen    | Shift PageUp   | Fn Shift Up   |
| Extend selection down one screen  | Shift PageDown   | Fn Shift Down   |
| Extend selection left one screen  | Alt shift PageUp   | Fn Shift ⌘ Up   |
| Extend selection right one screen | Alt shift PageDown   | Fn Shift ⌘ Down   |
| Select all                        | Ctrl A   | ⌘ A   |

## Data Editing Shortcuts

|                               |  |  |
|-------------------------------|--|---|
| Fill down from cell above     | Ctrl D   | Ctrl D  |
| Fill right from cell left     | Ctrl R   | Ctrl R  |
| Find and replace              | Ctrl F   | Ctrl F  |
| Show all constants            | F5 Alt S O   | <input type="checkbox"/>  |
| Highlight cells with comments | F5 Alt S C   | <input type="checkbox"/>  |

## Data Editing (inside cell) Shortcuts

|   | Windows                    | Mac                       |
|---|----------------------------|---------------------------|
| Edit the active cell (edit mode)                                | F2                         | F2                        |
| While editing cell, allow use of arrow keys to create reference | F2                         | F2                        |
| Confirm change and get out of cell                              | Enter                      | Enter                     |
| Cancel a cell entry and get out of cell                         | Esc                        | Esc                       |
| Insert line break within cell                                   | Alt Enter                  | Option Enter              |
| Highlight within a cell   | Shift Left/Right           | Shift Left/Right          |
| Highlight contiguous items                                      | Ctrl Shift Left/Right      | Ctrl Shift Left/Right     |
| Jump to beginning of cell contents                              | Home                       | Fn Left                   |
| Jump to end of cell contents                                    | End                        | Fn Right                  |
| Delete character to left  | Backspace                  | Delete                    |
| Delete character to right                                       | Delete                     | Fn Delete                 |
| Accept autocomplete suggestion                                  | Tab                        | Tab                       |
| Referencing a cell from another worksheet                       | Ctrl PageUp/Down<br>Arrows | Ctrl Fn Down/Up<br>Arrows |

## Other Shortcuts

|  | Windows | Mac    |
|--|---------|--------|
| Enter date   | Ctrl ;  | Ctrl ; |
| Enter time   | Ctrl :  | Ctrl : |
| Show formula/show values (key to the left of 1)  | Ctrl `  | Ctrl ` |
| Select cells which refer to the active cell (useful before deleting a cell in a worksheet) | Ctrl ]  | Ctrl ] |
| Drives menu bar  | Alt     | ⌘      |
| Next open program  | Alt Tab | ⌘ Tab  |
| Autosum  | Alt =   | ⌘ T    |

## Reasons to use Excel Shortcuts

To be more productive, faster, and more efficient **when building financial models** or performing financial analysis it's important to know the main keyboard shortcuts in Excel. These are critical for careers in investment banking, equity research, FP&A, finance, accounting, and more.

The first thing you'll do if you're hired as an investment banking analyst is take a series of intense **Excel training courses**. Your mouse will be taken away and you'll be expected to learn financial modeling with only keyboard shortcuts. If you follow our tips and tricks below you'll be able to master these shortcuts on Windows or Mac operating systems.

You may also want to check out our section on **Excel formulas** or the **Excel formulas cheat sheet**. Excel is quite robust, meaning there is a lot of different tools that can be utilized within it, and therefore there are **many skills** one can practice and hone.

## Free Excel Course

If you want to learn these keyboard shortcuts with your own personal online instructor, check our CFI's **Free Excel Crash Course!** You'll receive step by step instructions and demonstrations on how to avoid the mouse and only use your keyboard.

The screenshot shows the course interface for "Excel Crash Course - Spreadsheet Formulas for Corporate Finance". The left sidebar contains a navigation menu with sections like "Introduction", "Excel Layout", "Excel Settings", "Basic Financial Analysis Setup", "Advanced Financial Analysis Setup", "Objectives", "Structure, organization, duplicate sheets, grouping", "Anchoring cells, grouping", "Naming cells, using precedents & dependents, auditing", "Go to special, conditional formatting", "Dynamic links & headers", and "Functions & Formulas". The main area shows a video player with a play button overlay. The video content displays an Excel spreadsheet with a financial model, including columns for 2014 and 2015A, and rows for various financial metrics like Revenue, COGS, Operating Profit, EBITDA, EBIT, EBT, Taxes, and Net Income. A "Play" button is visible at the bottom left of the video player.

PART 02

# Basic Excel Formulas



# Basic Excel Formulas Guide

To learn more, please check out our online courses



Mastering the basic Excel formulas is critical for beginners to become highly proficient in financial analysis. **Microsoft Excel** is considered the industry standard piece of software in data analysis. Microsoft's spreadsheet program also happens to be one of the most preferred software by **investment bankers** and **financial analyst** in data processing, **financial modeling**, and presentation. This guide will provide an overview and list of basic Excel functions. Once you've mastered this list, move on to CFI's **advanced Excel formulas guide!**

## Basic Terms in Excel

### 1. Formulas

In Excel, a formula is an expression that operates on values in a range of cells or a cell. For example, `=A1+A2+A3`, which finds the sum of the range of values from cell A1 to Cell A3.

### 2. Functions

Functions are predefined formulas in Excel. They eliminate laborious manual entry of formulas while giving them human-friendly names. For example: `=SUM(A1:A3)`. The function sums all the values from A1 to A3.

## Five Time-saving Ways to Insert Data into Excel

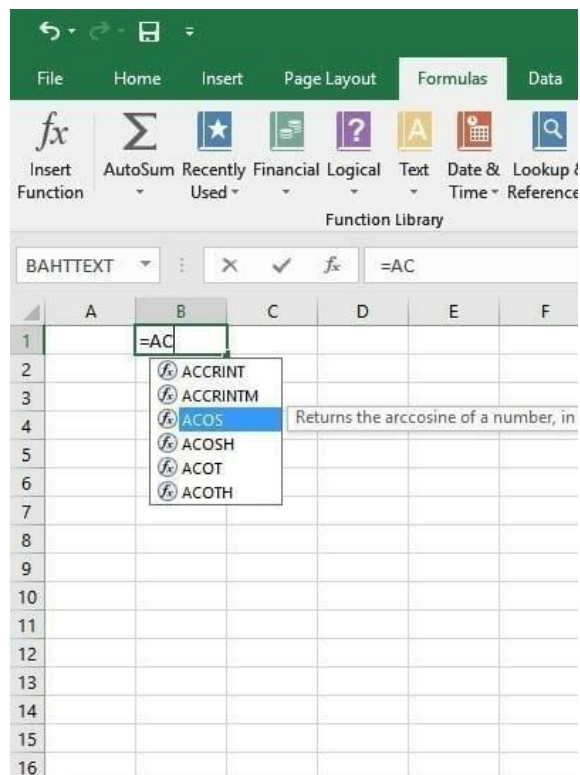
When analyzing data, there are five common ways of inserting basic Excel formulas. Each strategy, however, comes with an advantage over the other. Therefore, before diving further into the main formulas, we'll clarify those methods, so you can create your preferred workflow earlier on.

### 1. Simple insertion: Typing a formula inside the cell

Typing a formula in a cell or the formula bar is the most straightforward method of inserting basic Excel formulas. The process usually starts by typing an equal sign, followed by the name of the function.

Excel is quite intelligent in that when you start typing the name of the function, a pop-up function hint will show. It's from this list you'll select your preference. However, don't press the Enter key. Instead, press the Tab key so that you can continue to insert other options. Otherwise, you may find yourself with an invalid name error, often as '#NAME?'. To fix it, just re-select the cell, and go to the formula bar to complete your function.

Note that pressing F2 while hovered over a cell allows you to edit the cells' formula.

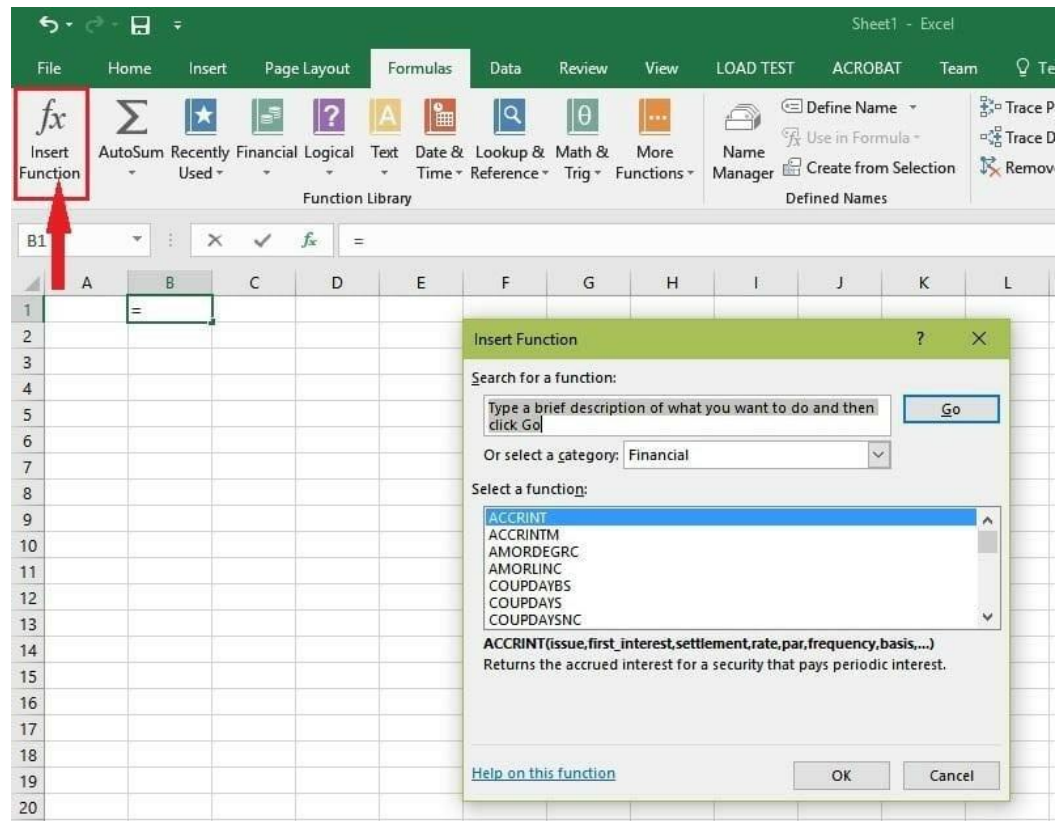




## 2. Using Insert Function Option from Formulas Tab

If you want full control of your functions insertion, using the Excel Insert Function dialogue box is all you ever need. To achieve this, go to the Formulas tab and select the first menu labeled Insert Function. The dialogue box will contain all functions you need to complete your analysis.

The Excel shortcut to insert a function is ALT + M + F.



### 3. Selecting a Formula from One of the Groups in Formula Tab

The option is for those who want to delve into their favorite functions quickly. To find this menu, navigate to the Formulas tab and select your preferred group. Click to show sub-menu filled with a list of functions. From there, you can select your preference. However, if you find your preferred group is not on the tab, click on the More Functions option – probably it's just hidden there.

The Excel formula shortcuts are the following:

Recently used: ALT + M + R

Financial: ALT + M + I

Logical: ALT + M + L

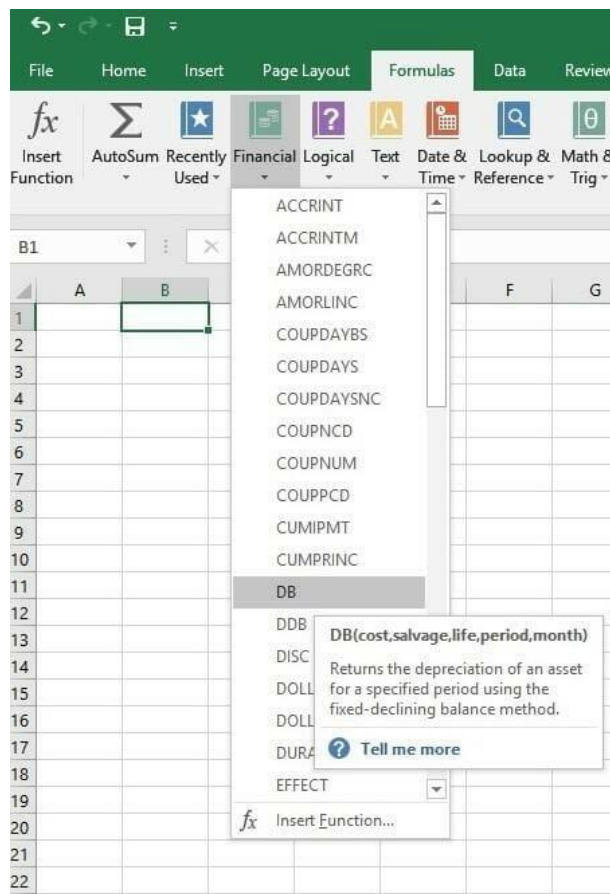
Text: ALT + M + T

Date & Time: ALT + M + E

Lookup & Reference: ALT + M + O

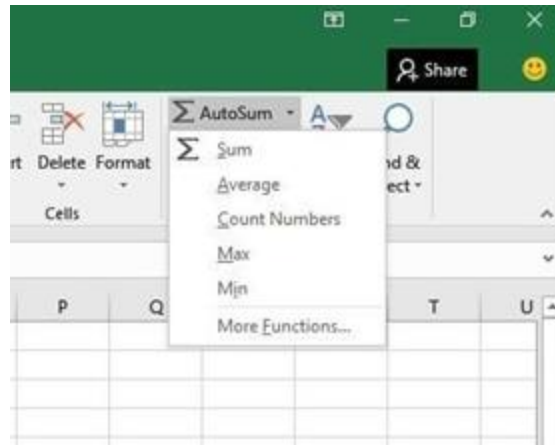
Math & Trig: ALT + M + G

More Function: ALT + M + Q



#### 4. Using AutoSum Option

For quick and everyday tasks, AutoSum is your go-to option. So, navigate to the Home tab, in the far-right corner, click the AutoSum option. Then click the caret to show other hidden formulas. This option is also available in the Formulas tab first option after the Insert Function option.



Alternatively, the Autosum Excel function can be accessed by typing ALT + the = sign in a spreadsheet and it will automatically create a formula to sum all the numbers in a continuous range.

**Step 1:** Place the cursor below the column of numbers you want to sum (or to the left of the row of numbers you want to sum).

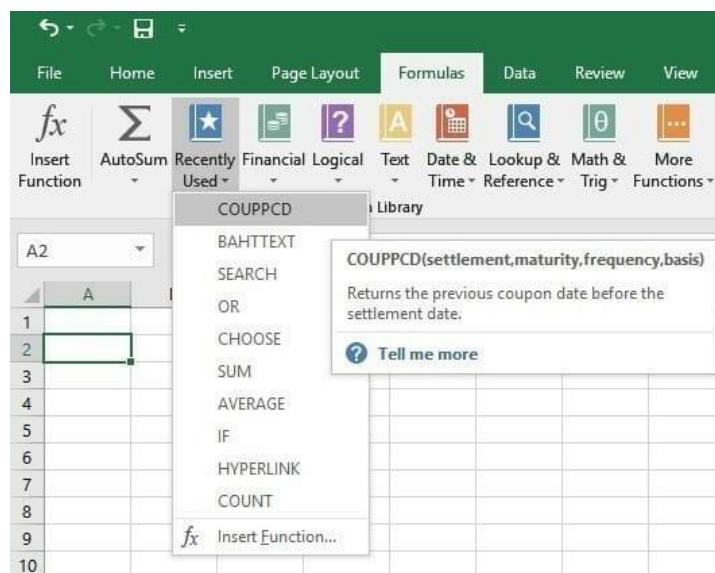
**Step 2:** Hold down the Alt key and then press the equals = sign while still holding Alt.

**Step 3:** Press Enter.

|    | A | B | C                            | D | E | F |
|----|---|---|------------------------------|---|---|---|
| 1  |   |   |                              |   |   |   |
| 2  |   |   | 10                           |   |   |   |
| 3  |   |   | 35                           |   |   |   |
| 4  |   |   | -3                           |   |   |   |
| 5  |   |   | 21                           |   |   |   |
| 6  |   |   | 8                            |   |   |   |
| 7  |   |   | 978                          |   |   |   |
| 8  |   |   | -50                          |   |   |   |
| 9  |   |   | 12                           |   |   |   |
| 10 |   |   | <b>Alt =</b> =SUM(C2:C9)     |   |   |   |
| 11 |   |   | SUM(number1, [number2], ...) |   |   |   |

## 5. Quick Insert: Use Recently Used Tabs

If you find re-typing your most recent formula a monotonous task, then use the Recently Used menu. It's on the Formulas tab, a third menu option just next to AutoSum.



## Seven Basic Excel Formulas For Your Workflow

Since you're now able to insert your preferred formulas and function correctly, let's check some fundamental Excel functions to get you started.

### 1. SUM

The SUM function is the first must-know formula in Excel. It usually aggregates values from a selection of columns or rows from your selected range.

**=SUM(number1, [number2], ...)**

Example:

**=SUM(B2:G2)** – A simple selection that sums the values of a row.

**=SUM(A2:A8)** – A simple selection that sums the values of a column.

**=SUM(A2:A7, A9, A12:A15)** – A sophisticated collection that sums values from range A2 to A7, skips A8, adds A9, jumps A10 and A11, then finally adds from A12 to A15.

**=SUM(A2:A8)/20** – Shows you can also turn your function into a formula.

The screenshot shows an Excel spreadsheet with the following data:

|    | A           | B           | C | D | E |
|----|-------------|-------------|---|---|---|
| 1  | Names       | Population  |   |   |   |
| 2  | Poland      | 694668023   |   |   |   |
| 3  | Haiti       | 4124085191  |   |   |   |
| 4  | Ireland     | 4376641145  |   |   |   |
| 5  | Kuwait      | 9624154387  |   |   |   |
| 6  | Belarus     | 6276743422  |   |   |   |
| 7  | Saint Bartl | 3042522977  |   |   |   |
| 8  | Singapore   | 4498405895  |   |   |   |
| 9  | South Afri  | 8205796639  |   |   |   |
| 10 | Sint Maart  | 7599707279  |   |   |   |
| 11 | Greenland   | 1202372341  |   |   |   |
| 12 |             | 49645097299 |   |   |   |
| 13 |             |             |   |   |   |

The formula bar at the top shows the formula **=SUM(B2:B11)** being entered into cell B12. The spreadsheet also shows a 'Formula Bar' label above the formula bar.

## 2. AVERAGE

The AVERAGE function should remind you of simple averages of data such as the average number of shareholders in a given shareholding pool.

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

Example:

**=AVERAGE(A1:A10)** – Shows a simple average, also similar to (SUM(A1: A10)/9)

|    | A                | B                                | C | D |
|----|------------------|----------------------------------|---|---|
| 1  | <b>Names</b>     | <b>Population</b>                |   |   |
| 2  | Poland           | 694668023                        |   |   |
| 3  | Haiti            | 4124085191                       |   |   |
| 4  | Ireland          | 4376641145                       |   |   |
| 5  | Kuwait           | 9624154387                       |   |   |
| 6  | Belarus          | 6276743422                       |   |   |
| 7  | Saint Barthélemy | 3042522977                       |   |   |
| 8  | Singapore        | 4498405895                       |   |   |
| 9  | South Africa     | 8205796639                       |   |   |
| 10 | Sint Maarten     | 7599707279                       |   |   |
| 11 | Greenland        | 1202372341                       |   |   |
| 12 | <b>Average</b>   | <b>=AVERAGE(B2:B11)</b>          |   |   |
| 13 |                  | AVERAGE(number1, [number2], ...) |   |   |
| 14 |                  |                                  |   |   |

### 3. COUNT

The COUNT function counts all cells in a given range that contains only numeric values.

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

Example:

**COUNT(A:A)** – Counts all values that are numerical in A column. However, it doesn't use the same formula to count rows.

**COUNT(A1:C1)** – Now it can count rows.

The screenshot shows an Excel spreadsheet with the following data:

|    | A                | B                 | C                 | D         |
|----|------------------|-------------------|-------------------|-----------|
| 1  | <b>Names</b>     | <b>Population</b> |                   |           |
| 2  | Poland           | 694668023         |                   |           |
| 3  | Haiti            | 4124085191        |                   |           |
| 4  | America          | 7/12/2050         |                   |           |
| 5  | Ireland          | 4376641145        |                   |           |
| 6  | Kuwait           | 9624154387        |                   |           |
| 7  | China            |                   |                   |           |
| 8  | Belarus          | 6276743422        |                   |           |
| 9  | Saint Barthélemy | 3042522977        |                   |           |
| 10 | Singapore        | 4498405895        |                   |           |
| 11 | South Africa     | 8205796639        |                   |           |
| 12 | Sint Maarten     | 7599707279        |                   |           |
| 13 | Turkey           | Text              |                   |           |
| 14 | Greenland        | 1202372341        |                   |           |
| 15 | <b>Count</b>     | <b>11</b>         | <b>Total Cell</b> | <b>13</b> |
| 16 |                  |                   |                   |           |

#### 4. COUNTA

Like the COUNT function, COUNTA counts all cells in a given range. However, it counts all cells regardless of type. That is, unlike COUNT that relies on only numerics, it also counts dates, times, strings, logical values, errors, empty string, or text.

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

##### Example:

**COUNTA(A:A)** – Counts all cells in column A regardless of type. However, like COUNT, you can't use the same formula to count rows.

|    | A                | B                 | C                  | D         |
|----|------------------|-------------------|--------------------|-----------|
| 1  | <b>Country</b>   | <b>Population</b> |                    |           |
| 2  | Poland           | 694668023         |                    |           |
| 3  | Haiti            | 4124085191        |                    |           |
| 4  | America          | 7/12/2050         |                    |           |
| 5  | Ireland          | 4376641145        |                    |           |
| 6  | Kuwait           | 9624154387        |                    |           |
| 7  | China            |                   |                    |           |
| 8  | Belarus          | 6276743422        |                    |           |
| 9  | Saint Barthélemy | 3042522977        |                    |           |
| 10 | Singapore        | 4498405895        |                    |           |
| 11 | South Africa     | 8205796639        |                    |           |
| 12 | Sint Maarten     | 7599707279        |                    |           |
| 13 | Turkey           | Text              |                    |           |
| 14 | Greenland        | 1202372341        |                    |           |
| 15 | <b>COUNTA</b>    | <b>12</b>         | <b>Total Cells</b> | <b>13</b> |
| 16 |                  |                   |                    |           |



## 5. IF

The IF function is often used when you want to sort your data according to a given logic. The best part of the IF formula is that you can embed formulas and function in it.

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

Example:

**=IF(C2<D3, 'TRUE,' FALSE')** – Checks if the value at C3 is less than the value at D3. If the logic is true, let the cell value be TRUE, else, FALSE

**=IF(SUM(C1:C10) > SUM(D1:D10), SUM(C1:C10), SUM(D1:D10))** – An example of a complex IF logic. First, it sums C1 to C10 and D1 to D10, then it compares the sum. If the sum of C1 to C10 is greater than SUM of D1 to D10, then it makes the value of a cell equal to the sum of C1 to C10. Otherwise, it makes it the SUM of D1 to D10.

|    | A                | B                 | C                            | D                         |
|----|------------------|-------------------|------------------------------|---------------------------|
| 1  | <b>Country</b>   | <b>Population</b> | <b>Comparison Population</b> | <b>IF is Greater than</b> |
| 2  | Poland           | 694668023         | 2084004069                   | FALSE                     |
| 3  | Haiti            | 4124085191        | 589155027                    | TRUE                      |
| 4  | Ireland          | 4376641145        | 625234449                    | TRUE                      |
| 5  | Kuwait           | 9624154387        | 9624243417                   | FALSE                     |
| 6  | Belarus          | 6276743422        | 18830230266                  | FALSE                     |
| 7  | Saint Barthélemy | 3042522977        | 434646140                    | TRUE                      |
| 8  | Singapore        | 4498405895        | 642629414                    | TRUE                      |
| 9  | South Africa     | 8205796639        | 24617389917                  | FALSE                     |
| 10 | Sint Maarten     | 7599707279        | 7599796309                   | FALSE                     |
| 11 | Greenland        | 1202372341        | 1202461371                   | FALSE                     |

## 6. TRIM

The TRIM function makes sure your functions do not return errors due to unruly spaces. It ensures that all empty spaces are eliminated. Unlike other functions that can operate on a range of cells, TRIM only operates on a single cell. Therefore, it comes with the downside of adding duplicated data in your spreadsheet.

**=TRIM(text)**

Example:

**TRIM(A4)** – Removes empty spaces in the value in cell A4.

|   | A              | B           | C |
|---|----------------|-------------|---|
| 1 | <b>Country</b> | <b>TRIM</b> |   |
| 2 | 2 Poland       | 2 Poland    |   |
| 3 | Haiti 20       | Haiti 20    |   |
| 4 | Ireland 23     | Ireland 23  |   |
| 5 | Kuwait 7       | Kuwait 7    |   |
| 6 | Belarus 909    | Belarus 909 |   |
| 7 |                |             |   |

## 7. MAX & MIN

The MAX and MIN functions help in finding the maximum number and the minimum number in a pull of values.

**=MIN(number1, [number2], ...)**

Example:

**=MIN(B2:C11)** – Finds the minimum number between column B from B2 and column C from C2 to row 11 in both column B and C.

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

Example:

**=MAX(B2:C11)** – Similarly, it finds the maximum number between column B from B2 and column C from C2 to row 11 in both column B and C.

|    | A                | B                 | C           | D |
|----|------------------|-------------------|-------------|---|
| 1  | <b>Country</b>   | <b>Population</b> |             |   |
| 2  | Poland           | 694668023         |             |   |
| 3  | Haiti            | 4124085191        |             |   |
| 4  | Ireland          | 4376641145        |             |   |
| 5  | Kuwait           | 9624154387        |             |   |
| 6  | Belarus          | 6276743422        |             |   |
| 7  | Saint Barthélemy | 3042522977        |             |   |
| 8  | Singapore        | 4498405895        |             |   |
| 9  | South Africa     | 8205796639        |             |   |
| 10 | Sint Maarten     | 7599707279        |             |   |
| 11 | Greenland        | 1202372341        |             |   |
| 12 | <b>MAX</b>       | <b>9624154387</b> | MAX(B2:B11) |   |
| 13 | <b>MIN</b>       | <b>694668023</b>  | MIN(B2:B11) |   |

PART 03

# Advanced Excel Formulas



# Advanced Excel Formulas You Must Know

To learn more, please check out our online courses



Every **financial analyst** spends more time in Excel than they may care to admit. Based on years and years of experience, we have compiled the most important and advanced Excel formulas that every world-class financial analyst must know.

## 1. INDEX MATCH

Formula: `=INDEX(C3:E9,MATCH(B13,C3:C9,0),MATCH(B14,C3:E3,0))`

This is an advanced alternative to the VLOOKUP or HLOOKUP formulas (which several drawbacks and limitations). **INDEX MATCH** is a powerful combination of Excel formulas that will take your financial analysis and financial modeling to the next level.

INDEX returns the value of a cell in a table based on the column and row number.

MATCH returns the position of a cell in a row or column.

Here is an example of the INDEX and MATCH formulas combined together. In this example, we look up and return a person's height based on their name. Since name and height are both variables in the formula, we can change both of them!

|    | A | B | C   | D      | E      | F | G |
|----|---|---|---|--------|--------|---|---|
| 1  |   |   |   |        |        |   |   |
| 2  |   |   | 1   | 2      | 3      |   |   |
| 3  |   | 1 | Name  | Height | Weight |   |   |
| 4  |   | 2 | Sally   | 6.2    | 185    |   |   |
| 5  |   | 3 | Tom   | 5.9    | 170    |   |   |
| 6  |   | 4 | Kevin   | 5.8    | 175    |   |   |
| 7  |   | 5 | Amanda  | 5.5    | 145    |   |   |
| 8  |   | 6 | Carl  | 6.1    | 210    |   |   |
| 9  |   | 7 | Ned   | 6.0    | 180    |   |   |
| 10 |   |   |   |        |        |   |   |
| 11 |   |   |   |        |        |   |   |
| 12 |   |   | =INDEX(C3:E9,MATCH(B13,C3:C9,0),MATCH(B14,C3:E3,0)) |        |        |   |   |
| 13 |   |   | Kevin   |        |        |   |   |
| 14 |   |   | Height  |        |        |   |   |

For a step-by-step explanation or how to use this formula, please see our free guide on [how to use INDEX MATCH MATCH in Excel](#).

## 2. IF combined with AND / OR

Formula: =IF(AND(C2>=C4,C2<=C5),C6,C7)

Anyone who's spent a great deal of time in various [types of financial models](#) knows that nested IF formulas can be a nightmare. Combining IF with the AND or the OR function can be a great way to keep or formulas easier to audit and for other users to understand. In the example below, you will see how we used the individual functions in combination to create a more advanced formula.

For a detailed breakdown of how to perform this function in Excel please see our free guide on [how to use IF with AND / OR](#).

|    | A | B               | C                             | D  | E | F | G |  |
|----|---|-----------------|-------------------------------|----|---|---|---|--|
| 1  |   |                 |                               |    |   |   |   |  |
| 2  |   | Data Cell       | 150                           |    |   |   |   |  |
| 3  |   |                 |                               |    |   |   |   |  |
| 4  |   | Condition 1     | 100                           | >= |   |   |   |  |
| 5  |   | Condition 2     | 999                           | <= |   |   |   |  |
| 6  |   | Result if true  | 100                           |    |   |   |   |  |
| 7  |   | Result if fales | 0                             |    |   |   |   |  |
| 8  |   |                 |                               |    |   |   |   |  |
| 9  |   | Live Formula    | =IF(AND(C2>=C4,C2<=C5),C6,C7) |    |   |   |   |  |
| 10 |   |                 |                               |    |   |   |   |  |
| 11 |   |                 |                               |    |   |   |   |  |

### 3. OFFSET combined with SUM or AVERAGE

Formula: =SUM(B4:OFFSET(B4,0,E2-1))

The **OFFSET function** on its own is not particularly advanced, but when we combine it with other functions like SUM or AVERAGE we can create a pretty sophisticated formula. Suppose you want to create a dynamic function that can sum a variable number of cells. With the regular SUM formula, you are limited to a static calculation, but by adding OFFSET you can have the cell reference move around.

How it works. To make this formula work we substitute ending reference cell of the SUM function with the OFFSET function. This makes the formula dynamic and cell referenced as E2 is where you can tell Excel how many consecutive cells you want to add up. Now we've got some advanced Excel formulas!

Below is a screenshot of this slightly more sophisticated formula in action.

|    | A | B                      | C | D | E                          | F       | G     | H |
|----|---|------------------------|---|---|----------------------------|---------|-------|---|
| 1  |   |                        |   |   |                            |         |       |   |
| 2  |   | Sum this many numbers: |   |   | 3                          |         |       |   |
| 3  |   |                        |   |   |                            |         |       |   |
| 4  |   | 5                      | 8 | 2 | 9                          | 6       | 4     |   |
| 5  |   |                        |   |   |                            |         |       |   |
| 6  |   | Solution               |   |   |                            |         |       |   |
| 7  |   |                        |   |   | =SUM(B4:OFFSET(B4,0,E2-1)) | Formula |       |   |
| 8  |   |                        |   |   |                            | 15      | Value |   |
| 9  |   |                        |   |   |                            |         |       |   |
| 10 |   |                        |   |   |                            |         |       |   |
| 11 |   |                        |   |   |                            |         |       |   |

As you see, the SUM formula starts in cell B4, but it ends with a variable, which is the OFFSET formula starting at B4 and continuing by the value in E2 ("3") minus one. This moves the end of the sum formula over 2 cells, summing 3 years of data (including the starting point). As you can see in cell F7, the sum of cells B4:D4 is 15 which is what the offset and sum formula gives us.

[Learn how to build this formula step-by-step in our advanced Excel course.](#)

#### 4. CHOOSE

Formula: =CHOOSE(choice, option1, option2, option3)

The **CHOOSE function** is great for scenario analysis in financial modeling. It allows you to pick between a specific number of options, and return the "choice" that you've selected. For example, imagine you have three different assumptions for revenue growth next year: 5%, 12% and 18%. Using the CHOOSE formula you can return 12% if you tell Excel you want choice #2.

Read more about [scenario analysis in Excel](#).

|    | A | B            | C        | D                    | E | F | G |
|----|---|--------------|----------|----------------------|---|---|---|
| 1  |   |              |          |                      |   |   |   |
| 2  |   |              |          |                      |   |   |   |
| 3  |   |              | Option 1 | 5%                   |   |   |   |
| 4  |   |              | Option 2 | 12%                  |   |   |   |
| 5  |   |              | Option 3 | 18%                  |   |   |   |
| 6  |   |              |          |                      |   |   |   |
| 7  |   | Selection -> | 2        | =CHOOSE(C7,D3,D4,D5) |   |   |   |
| 8  |   |              |          |                      |   |   |   |
| 9  |   |              |          |                      |   |   |   |
| 10 |   |              |          |                      |   |   |   |
| 11 |   |              |          |                      |   |   |   |

To see a video demonstration, check out our [Advanced Excel Formulas Course](#).



## 5. XNPV and XIRR

Formula: =XNPV(discount rate, cash flows, dates)

If you're an analyst working in [investment banking](#), equity research, or financial planning & analysis (FP&A), or any other area of corporate finance that requires discounting cash flows then these formulas are a lifesaver!

Simply put, XNPV and XIRR allow you to apply specific dates to each individual cash flow that's being discounted. The problem with Excel's basic NPV and IRR formulas is that they assume the time periods between cash flow are equal. Routinely, as an analyst, you'll have situations where cash flows are not timed evenly, and this formula is how you fix that.

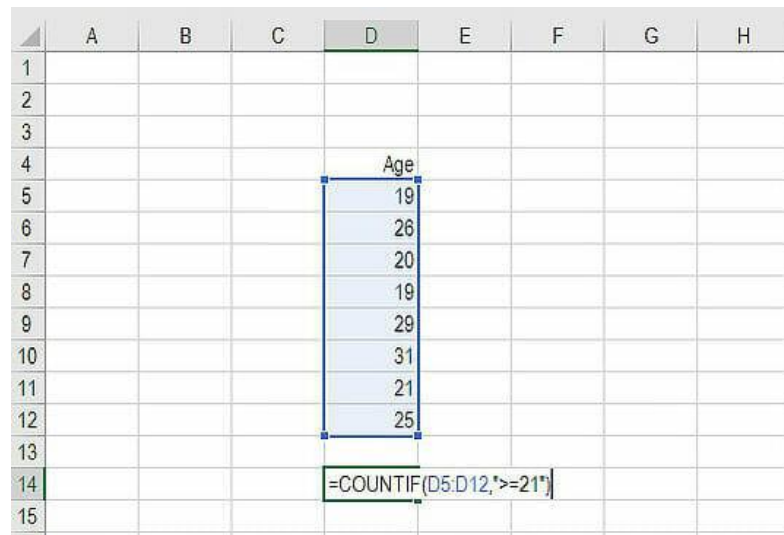
|    | A | B                    | C | D                     | E          | F         | G          | H        | I          |
|----|---|----------------------|---|-----------------------|------------|-----------|------------|----------|------------|
| 1  |   |                      |   |                       |            |           |            |          |            |
| 2  |   |                      |   |                       |            |           |            |          |            |
| 3  |   | <b>Dates</b>         |   | 5/18/2018             | 12/31/2018 | 9/12/2019 | 12/25/2019 | 5/8/2020 | 12/31/2020 |
| 4  |   |                      |   |                       |            |           |            |          |            |
| 5  |   | <b>Cash Flows</b>    |   | 1,000                 | 1,000      | 1,000     | 1,000      | 1,000    | 1,000      |
| 6  |   |                      |   |                       |            |           |            |          |            |
| 7  |   | <b>Discount Rate</b> |   | 10.0%                 |            |           |            |          |            |
| 8  |   |                      |   |                       |            |           |            |          |            |
| 9  |   | <b>XNPV</b>          |   | =XNPV(D7,D5:I5,D3:I3) | Formula    |           |            |          |            |
| 10 |   |                      |   | 5,289                 | Value      |           |            |          |            |
| 11 |   |                      |   |                       |            |           |            |          |            |

For a more detailed breakdown, see our [free IRR vs XIRR formulas guide](#) as well as our [XNPV guide](#).

## 6. SUMIF and COUNTIF

Formula: =COUNTIF(D5:D12,">=21")

These two advanced formulas are great uses of conditional functions. SUMIF adds all cells that meet certain criteria, and COUNTIF counts all cells that meet certain criteria. For example, imagine you want to count all cells that are greater than or equal to 21 (the legal drinking age in the U.S.) to find out how many bottles of champagne you need for a client event. You can use COUNTIF as an advanced solution, as shown in the screenshot below.



|    | A | B | C | D                       | E | F | G | H |
|----|---|---|---|-------------------------|---|---|---|---|
| 1  |   |   |   |                         |   |   |   |   |
| 2  |   |   |   |                         |   |   |   |   |
| 3  |   |   |   |                         |   |   |   |   |
| 4  |   |   |   | Age                     |   |   |   |   |
| 5  |   |   |   | 19                      |   |   |   |   |
| 6  |   |   |   | 26                      |   |   |   |   |
| 7  |   |   |   | 20                      |   |   |   |   |
| 8  |   |   |   | 19                      |   |   |   |   |
| 9  |   |   |   | 29                      |   |   |   |   |
| 10 |   |   |   | 31                      |   |   |   |   |
| 11 |   |   |   | 21                      |   |   |   |   |
| 12 |   |   |   | 25                      |   |   |   |   |
| 13 |   |   |   |                         |   |   |   |   |
| 14 |   |   |   | =COUNTIF(D5:D12,">=21") |   |   |   |   |
| 15 |   |   |   |                         |   |   |   |   |

In our [advanced Excel course](#) we break these formulas down in even more detail.

## 7. PMT and IPMT

Formula: =PMT(interest rate, # of periods, present value)

If you work in **commercial banking**, real estate, **FP&A** or any financial analyst position that deals with debt schedules, you'll want to understand these two detailed formulas.

The PMT formula gives you the value of equal payments over the life of a loan. You can use it in conjunction with IPMT (which tells you the interest payments for the same type of loan) then separate principal and interest payments.

Here is an example of how to use the PMT function to get the monthly mortgage payment for a \$1 million mortgage at 5% for 30 years.

|    | A | B           | C                 | D       | E |
|----|---|-------------|-------------------|---------|---|
| 1  |   |             |                   |         |   |
| 2  |   |             |                   |         |   |
| 3  |   | Rate        | 5.0%              |         |   |
| 4  |   | # Perids    | 30                |         |   |
| 5  |   | Loan Value  | 1,000,000         |         |   |
| 6  |   |             |                   |         |   |
| 7  |   | PMT         | =PMT(C3,C4,C5,,1) | Formula |   |
| 8  |   | PMT         | 61,954            | Value   |   |
| 9  |   | Monthly PMT | 5,163             |         |   |
| 10 |   |             |                   |         |   |
| 11 |   |             |                   |         |   |

## 8. LEN and TRIM

Formulas: =LEN(text) and =TRIM(text)

These are a little less common, but certainly very sophisticated formulas. These applications are great for **financial analysts** that need to organize and manipulate large amounts of data. Unfortunately, the data we get is not always perfectly organized and sometimes there can be issues like extra spaces at the beginning or end of cells

In the example below, you can see how the TRIM formula cleans up the Excel data.

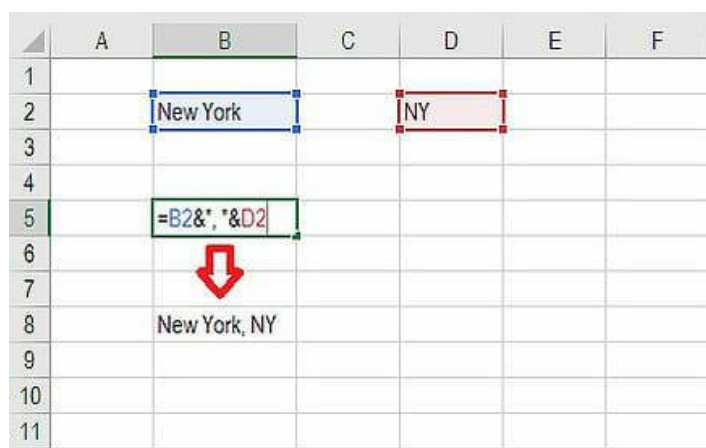
|    | A | B                       | C | D | E                       | F | G |
|----|---|-------------------------|---|---|-------------------------|---|---|
| 1  |   |                         |   |   |                         |   |   |
| 2  |   |                         |   |   |                         |   |   |
| 3  |   | No Extra spaces         |   |   |                         |   |   |
| 4  |   | Example of extra spaces |   |   | =TRIM(B4)               |   |   |
| 5  |   |                         |   |   | ↓                       |   |   |
| 6  |   |                         |   |   |                         |   |   |
| 7  |   | Example of extra spaces |   |   | Example of extra spaces |   |   |
| 8  |   |                         |   |   |                         |   |   |
| 9  |   |                         |   |   |                         |   |   |
| 10 |   |                         |   |   |                         |   |   |

## 9. CONCATENATE

Formula: =A1&" more text"

Concatenate is not really a function on its own, it's just an innovative way of joining information from different cells, and making worksheets more dynamic. This is a very powerful tool for financial analysts performing financial modeling (see our [free financial modeling guide](#) to learn more).

In the example below, you can see how the text "New York" plus ", " is joined with "NY" to create "New York, NY". This allows you to create dynamic headers and labels in worksheets. Now, instead of updating cell B8 directly, you can update cells B2 and D2 independently. With a large dataset this is a valuable skill to have at your disposal.






|    | A | B            | C | D  | E | F |
|----|---|--------------|---|----|---|---|
| 1  |   |              |   |    |   |   |
| 2  |   | New York     |   | NY |   |   |
| 3  |   |              |   |    |   |   |
| 4  |   |              |   |    |   |   |
| 5  |   | =B2&","&D2   |   |    |   |   |
| 6  |   |              |   |    |   |   |
| 7  |   |              |   |    |   |   |
| 8  |   | New York, NY |   |    |   |   |
| 9  |   |              |   |    |   |   |
| 10 |   |              |   |    |   |   |
| 11 |   |              |   |    |   |   |

## 10. CELL, LEFT, MID and RIGHT functions

These advanced Excel functions can be combined to create some very advanced and complex formulas to use. The CELL function can return a variety of information about the contents of a cell (its name, location, row, column, and more). The LEFT function can return text from the beginning of a cell (left to right), MID returns text from any start point of the cell (left to right), and RIGHT returns text from the end of the cell (right to left).

Below is an illustration of these three formulas in action.

To see how these can be combined in a powerful way with the CELL function, we break it down for you step by step in our [advanced Excel formulas class](#).

|    | A | B            | C | D            | E   | F    |
|----|---|--------------|---|--------------|---|------|
| 1  |   |              |   |              |   |      |
| 2  |   |              |   |              |   |      |
| 3  |   | New York, NY |   | =LEFT(B3,3)  |  | New  |
| 4  |   |              |   |              |   |      |
| 5  |   |              |   | =MID(B3,5,4) |  | York |
| 6  |   |              |   |              |   |      |
| 7  |   |              |   | =RIGHT(B3,2) |  | NY   |
| 8  |   |              |   |              |   |      |
| 9  |   |              |   |              |   |      |
| 10 |   |              |   |              |   |      |
| 11 |   |              |   |              |   |      |

## PART 04

# Most Useful Excel functions For Financial Modeling



01

02

03

# Excel Functions List

To learn more, please check out our online courses



## Date and Time

**DATE** Create a valid date from year, month, and day

year

month

day

### What is the DATE Function?

The DATE Function in Excel is categorized under **Date/Time Functions**. It is the main function used to calculate dates in Excel. The DATE function is very useful for **financial analysts** because **financial modeling** requires specific time periods. For example, an analyst can use the DATE function in Excel in their financial model to dynamically link the year, month and day from different cells into one function.

The DATE function is also useful when providing dates as inputs for other functions like SUMIFS or COUNTIFS since you can easily assemble a date using year, month, and day values that come from a cell reference or formula result.

### Formula

**=DATE(year,month,day)**

The DAYS function includes the following arguments:

1. **Year** – It is a required argument. The value of the year argument can include one to four digits. Excel interprets the year argument according to the date system used by the local computer. By default, Microsoft Excel for Windows uses the 1900 date system, which means the first date is January 1, 1900.
2. **Month** – It is a required argument. It can be a positive or negative integer representing the month of the year from 1 to 12 (January to December).
  - Excel will add the number of months to the first month of the specified year. For example, DATE(2017,14,2) returns the serial number representing February 2, 2018.
  - When the month is less than or equal to zero, Excel will subtract the absolute value of month plus 1 from the first month of the specified year.

For example, DATE(2016,-3,2) returns the serial number representing September 2, 2015.



3. **Day** – It is a required argument. It can be a positive or negative integer representing day of a month from 1 to 31.
- When day is greater than the number of days in the specified month, day adds that number of days to the first day of the month. For example, DATE(2016,1,35) returns the serial number representing February 4, 2016.
  - When day is less than 1, this function will subtract the value of the number of days, plus one, from the first day of the month specified. For example, DATE(2016,1,-15) returns the serial number representing December 16, 2015.

### How to use the DATE Function in Excel?

The Date function is a built-in function that can be used as a worksheet function in Excel. To understand the uses of this function, let's consider few examples:

#### Example 1

Let's see how this function works using the different examples below:

| Formula used                           | Result           | Remarks   |
|--|------------------|---|
| DATE(YEAR(TODAY()), MONTH(TODAY()), 1) | January 11, 2017 | Returns the first day of the current year and month |
| DATE(2017, 5, 20)-15                   | May 5, 2017      | Subtracts 15 days from May 20, 2017                 |

## Example 2

Suppose we need to calculate the percentage remaining in a year based on a given date. We can do so with a formula based on the YEARFRAC function.

|   | A | B                    |
|---|---|----------------------|
| 1 |   |                      |
| 2 |   | <b>DATE Function</b> |
| 3 |   |                      |
| 4 |   | <b>Data</b>          |
| 5 |   | 29/2/16              |
| 6 |   | 15/1/16              |
| 7 |   | 23/3/17              |
| 8 |   | 17/7/17              |
| 9 |   | 21/11/17             |

The formula to be used is:

| IRR                              |   |   |                      |                                  |                                  |   |   |
|----------------------------------|---|---|----------------------|----------------------------------|----------------------------------|---|---|
| =YEARFRAC(DATE(YEAR(B5),1,1),B5) |   |   |                      |                                  |                                  |   |   |
| A                                | B | C | D                    | E                                | F                                | G | H |
| 1                                |   |   |                      |                                  |                                  |   |   |
| 2                                |   |   | <b>DATE Function</b> |                                  |                                  |   |   |
| 3                                |   |   |                      |                                  |                                  |   |   |
| 4                                |   |   | <b>Data</b>          | <b>Formula Used</b>              | <b>Result</b>                    |   |   |
| 5                                |   |   | 29/2/16              | =YEARFRAC(DATE(YEAR(B5),1,1),B5) | =YEARFRAC(DATE(YEAR(B5),1,1),B5) |   |   |
| 6                                |   |   | 15/1/16              | =YEARFRAC(DATE(YEAR(B6),1,1),B6) | 4%                               |   |   |
| 7                                |   |   | 23/3/17              | =YEARFRAC(DATE(YEAR(B7),1,1),B7) | 23%                              |   |   |
| 8                                |   |   | 17/7/17              | =YEARFRAC(DATE(YEAR(B8),1,1),B8) | 54%                              |   |   |
| 9                                |   |   | 21/11/17             | =YEARFRAC(DATE(YEAR(B9),1,1),B9) | 89%                              |   |   |
| 10                               |   |   |                      |                                  |                                  |   |   |

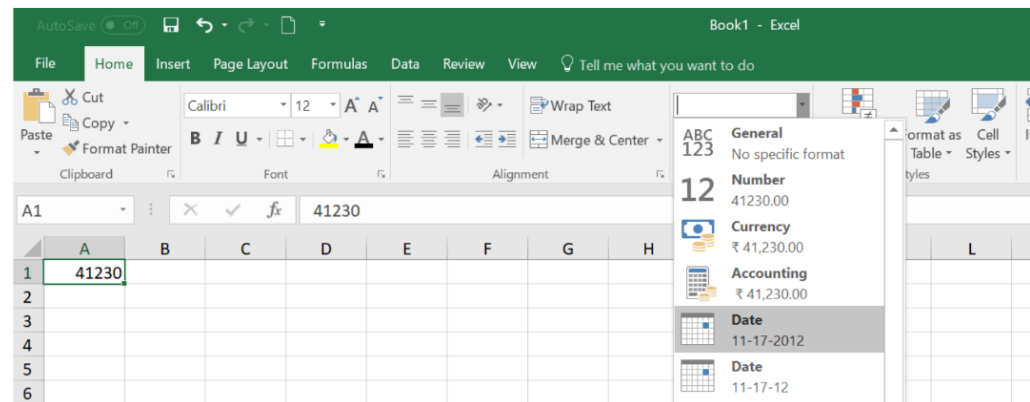
We get the result below:

| D5                               |   |   |                      |                                  |               |
|----------------------------------|---|---|----------------------|----------------------------------|---------------|
| =YEARFRAC(DATE(YEAR(B5),1,1),B5) |   |   |                      |                                  |               |
| A                                | B | C | D                    | E                                |               |
| 1                                |   |   |                      |                                  |               |
| 2                                |   |   | <b>DATE Function</b> |                                  |               |
| 3                                |   |   |                      |                                  |               |
| 4                                |   |   | <b>Data</b>          | <b>Formula Used</b>              | <b>Result</b> |
| 5                                |   |   | 29/2/16              | =YEARFRAC(DATE(YEAR(B5),1,1),B5) | 16%           |
| 6                                |   |   | 15/1/16              | =YEARFRAC(DATE(YEAR(B6),1,1),B6) | 4%            |
| 7                                |   |   | 23/3/17              | =YEARFRAC(DATE(YEAR(B7),1,1),B7) | 23%           |
| 8                                |   |   | 17/7/17              | =YEARFRAC(DATE(YEAR(B8),1,1),B8) | 54%           |
| 9                                |   |   | 21/11/17             | =YEARFRAC(DATE(YEAR(B9),1,1),B9) | 89%           |
| 10                               |   |   |                      |                                  |               |

## A few things to remember about the DATE function

1. #NUM! error – Occurs when the given year argument is  $< 0$  or  $\geq 10000$ .
2. #VALUE! Error – Occurs if any of the given argument is non-numeric.
3. Suppose while using this function you get a number such as 41230 instead of a date. Generally, it will occur due to the formatting of the cell. The function returned the correct value, but the cell is displaying the date serial number, instead of the formatted date.

Hence, we need to change the formatting of the cell to display a date. The easiest and quickest way to do this is to select the cell(s) to be formatted and then select the Date cell formatting option from the drop-down menu in the 'Number' group on the Home tab (of the Excel ribbon), as shown below:



**EOMONTH** Get the last day of the month in future or past months

start\_date

months

### What is the EOMONTH Function?

The EOMONTH Function is categorized under **DATE/TIME functions**. The function helps to calculate the last day of the month after adding a specified number of months to a date.

As a **financial analyst**, the EOMONTH Function becomes useful when we are calculating maturity dates for accounts payable or accounts payable that fall on the last day of the month. It also helps in calculating due dates that fall on the last day of the month. In financial analysis, we often analyze revenue generated by an organization. The function helps us do that in some cases.

### Formula

**=EOMONTH(start\_date, months)**

The EOMONTH function includes the following arguments:

1. **Start\_date** (required argument) – It is the initial date. We need to enter dates in date format either by using the DATE function or as results of other formulas or functions. For example, use DATE(2017,5,13) for May 13, 2017. This function will return errors if dates are entered as text.
2. **Months** (required argument) – It is the number of months before or after start\_date. A positive value for months yields a future date; a negative value produces a past date.

### How to use the EOMONTH Function in Excel?

To understand the uses of this function, let's consider few examples:

### Example 1

Let's see what results we get when we provide following data:

|    | A | B                       | C                   | D             | F |
|----|---|-------------------------|---------------------|---------------|---|
| 1  |   |                         |                     |               |   |
| 2  |   | <b>EOMONTH Function</b> |                     |               |   |
| 3  |   |                         |                     |               |   |
| 4  |   | <b>Data</b>             | <b>Formula used</b> | <b>Result</b> |   |
| 5  |   | 03/02/18                | =EOMONTH( A2, 9 )   | 12/31/18      |   |
| 6  |   | 04/15/16                | =EOMONTH( A3, -9 )  | 07/31/15      |   |
| 7  |   | 01/01/17                | =EOMONTH( A4, 0 )   | 01/31/17      |   |
| 8  |   | 01/01/17                | =EOMONTH( A5, 12 )  | 01/31/18      |   |
| 9  |   |                         | =EOMONTH( TODAY(),  | 12/31/18      |   |
| 10 |   |                         |                     |               |   |

In Row 2, the function added 9 to return the last day of December (9+3). In Row 3, the function subtracted 9 to return July 31, 2015. In Row 4, the function just returned January 31, 2017. In Row 5, the function added 12 months and returned the last day in January.

In Row 6, we used the function TODAY so EOMONTH took the date as of today that is November 18, 2017, and added 9 months to it to return August 31, 2018.

Remember that the EOMONTH function will return a serial date value. A serial date is how Excel stores dates internally and it represents the number of days since January 1, 1900.

## Example 2 – Using EOMONTH with SUMIF

Assume we are given the sales of different products in the following format:

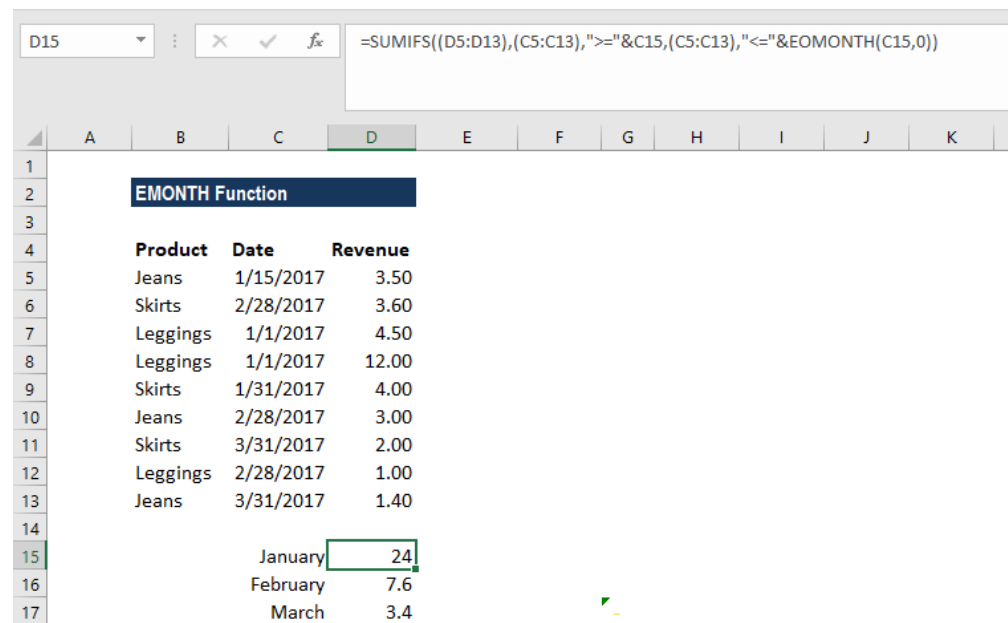
|    | A | B                      | C           | D              | E |
|----|---|------------------------|-------------|----------------|---|
| 1  |   |                        |             |                |   |
| 2  |   | <b>EMONTH Function</b> |             |                |   |
| 3  |   |                        |             |                |   |
| 4  |   | <b>Product</b>         | <b>Date</b> | <b>Revenue</b> |   |
| 5  |   | Jeans                  | 1/15/2017   | 3.50           |   |
| 6  |   | Skirts                 | 2/28/2017   | 3.60           |   |
| 7  |   | Leggings               | 1/1/2017    | 4.50           |   |
| 8  |   | Leggings               | 1/1/2017    | 12.00          |   |
| 9  |   | Skirts                 | 1/31/2017   | 4.00           |   |
| 10 |   | Jeans                  | 2/28/2017   | 3.00           |   |
| 11 |   | Skirts                 | 3/31/2017   | 2.00           |   |
| 12 |   | Leggings               | 2/28/2017   | 1.00           |   |
| 13 |   | Jeans                  | 3/31/2017   | 1.40           |   |

Now we wish to find out the total revenue that was achieved for the month of January, February, and March. The formula we will use is:

**SUMIFS((C3:C11),(B3:B11),">="&E4,(B3:B11),"<="&EOMONTH(E4,0))**

|    | A | B                      | C           | D              | E | F | G | H | I | J | K |
|----|---|------------------------|-------------|----------------|---|---|---|---|---|---|---|
| 1  |   |                        |             |                |   |   |   |   |   |   |   |
| 2  |   | <b>EMONTH Function</b> |             |                |   |   |   |   |   |   |   |
| 3  |   |                        |             |                |   |   |   |   |   |   |   |
| 4  |   | <b>Product</b>         | <b>Date</b> | <b>Revenue</b> |   |   |   |   |   |   |   |
| 5  |   | Jeans                  | 1/15/2017   | 3.50           |   |   |   |   |   |   |   |
| 6  |   | Skirts                 | 2/28/2017   | 3.60           |   |   |   |   |   |   |   |
| 7  |   | Leggings               | 1/1/2017    | 4.50           |   |   |   |   |   |   |   |
| 8  |   | Leggings               | 1/1/2017    | 12.00          |   |   |   |   |   |   |   |
| 9  |   | Skirts                 | 1/31/2017   | 4.00           |   |   |   |   |   |   |   |
| 10 |   | Jeans                  | 2/28/2017   | 3.00           |   |   |   |   |   |   |   |
| 11 |   | Skirts                 | 3/31/2017   | 2.00           |   |   |   |   |   |   |   |
| 12 |   | Leggings               | 2/28/2017   | 1.00           |   |   |   |   |   |   |   |
| 13 |   | Jeans                  | 3/31/2017   | 1.40           |   |   |   |   |   |   |   |

The result we get is:



The screenshot shows an Excel spreadsheet with the following data:

| Product  | Date      | Revenue |
|----------|-----------|---------|
| Jeans    | 1/15/2017 | 3.50    |
| Skirts   | 2/28/2017 | 3.60    |
| Leggings | 1/1/2017  | 4.50    |
| Leggings | 1/1/2017  | 12.00   |
| Skirts   | 1/31/2017 | 4.00    |
| Jeans    | 2/28/2017 | 3.00    |
| Skirts   | 3/31/2017 | 2.00    |
| Leggings | 2/28/2017 | 1.00    |
| Jeans    | 3/31/2017 | 1.40    |

| Month    | Revenue |
|----------|---------|
| January  | 24      |
| February | 7.6     |
| March    | 3.4     |

The SUMIFS formula added up all sales that occurred in January to give 24 (3.5+4.5+12+4) as the result for January. Similarly, we got the results for February and March.

What happened in this formula is that SUMIFS function totaled up the sales for the range given using two criteria:

1. One was to match dates greater than or equal to the first day of the month.
2. Second, to match dates less than or equal to the last day of the month.

So the formula worked in this way:

**=SUMIFS(revenue, (B3:B11), ">="&DATE(2017,1,1), (B3:B11), "<="&DATE(2017,1,31))**

Remember in column E, we must first type 1/1/2017 and then, using a custom format, change it to a custom date format ("mmmm") to display the month names. If we don't do this, the formula will not work properly.

Using concatenation with an ampersand (&) is necessary when building criteria that use a logical operator with a numeric value. Hence, we added that to the formula.

### Things to remember about the EOMONTH Function

1. #NUM! error – Occurs if either:
  1. The supplied start\_date is not a valid Excel date; or
  2. The supplied start\_date plus the value of the 'months' argument is not a valid Excel date.
2. #VALUE error – Occurs if any of the supplied arguments are non-numeric.
3. If we provide a decimal value for months, the EOMONTH function will only add the integer portion to start\_date.



**TODAY** Get the current date



### What is the TODAY Function?

The TODAY function is categorized under **Date and Time functions**. It will calculate and give the current date. It is updated continuously whenever a worksheet is changed or opened by a user. The function's purpose is to get today's date.

As a **financial analyst**, the TODAY function can be used when we wish to display the current date in a report. It is also helpful in calculating intervals. Suppose we are given a database of employees, we can use the function to calculate the age of employee as of today.

### Formula

**=TODAY()**

The TODAY function requires no arguments. However, it requires that you use empty parentheses ().

The function will continually update each time the worksheet is opened or recalculated, that is, each time a cell value is entered or changed. If the value doesn't change, we need to use F9 to force the worksheet to recalculate and update the value.

However, If we need a static date, we can enter the current date using the keyboard shortcut Ctrl + Shift + ;

### How to use the TODAY Function in Excel?

The TODAY function was introduced in Excel 2007 and is available in all subsequent Excel versions. To understand the uses of the function, let us consider an example:

### Example 1

Let's see how the function will behave when we give the following formulas:

| Formula              | Result   | Explanation  |
|----------------------|----------|--|
| =TODAY()             | 01/14/18 | The formula returned today's date.   |
| =TODAY()+15          | 01/29/18 | The formula added 15 days to today's date. As dates are stored in MS Excel as serial numbers, we can simply add +15. |
| =WORKDAY(TODAY(),30) | 02/23/18 | It added 30 work days to today's date.   |
| =MONTH(TODAY())      | 1        | It will return the current month of the year (1-12). As January is the current month, the result returned 1.         |

We get the results below:

|   | A | B                     | C | D | E | F |
|---|---|-----------------------|---|---|---|---|
| 1 |   |                       |   |   |   |   |
| 2 |   | <b>TODAY Function</b> |   |   |   |   |
| 3 |   |                       |   |   |   |   |
| 4 |   | 03/07/18              |   |   |   |   |
| 5 |   | 03/22/18              |   |   |   |   |
| 6 |   | 04/18/18              |   |   |   |   |
| 7 |   | 3                     |   |   |   |   |
| 8 |   |                       |   |   |   |   |

When we open this worksheet on a different date, the formulas will automatically update and give a different result.

## Example 2

Let's now understand how we can build a data validation rule for a date using the function. Suppose we wish to create a rule that allows only a date within the next 30 days, we can use data validation with a custom formula based on the AND, and TODAY functions.

Suppose we are given the following data:

|   | A | B                    | C               | D |
|---|---|----------------------|-----------------|---|
| 1 |   |                      |                 |   |
| 2 |   |                      |                 |   |
| 3 |   |                      |                 |   |
| 4 |   | <b>Checklist</b>     | <b>Due date</b> |   |
| 5 |   | Sending a quote      |                 |   |
| 6 |   | Follow-up            |                 |   |
| 7 |   | Signing the contract |                 |   |

Different users of this file will input dates for B5, B6, and B7. We will apply data validation to C5:C7. The formula to be applied would be:

If we try to input a date that is not within 30 days, we will get an error. Data validation rules are triggered when a user tries to add or change a cell value.

The TODAY function returns today's date. It will be recalculated on an ongoing basis. The AND function takes multiple logical expressions and will return TRUE only when all expressions return TRUE. In such case, we need to test two conditions:

$B3 > \text{TODAY}()$  – It checks that the date input by a user is greater than today.

$B3 \leq (\text{TODAY}() + 30)$  – It checks that the input is less than today plus 30 days.

Excel stores dates as sequential serial numbers so they can be used in calculations. By default, January 1, 1900 is serial number 1, and January 1, 2018 is serial number 43101 because it is 43,100 days after January 1, 1900. Hence, we can simply add the number of days as +30 to the TODAY function.

If both logical expressions return TRUE, the AND function returns TRUE and the data validation succeeds. If either expression returns FALSE, the validation fails.

**YEAR**

Get the year from a date

date

**What is the YEAR Function?**

The YEAR function is a **Date/Time function** that is used for calculating the year number from a given date. The function will return an integer that is a four-digit year corresponding to a specified date. For example, if we use this function on a date such as 12/12/2017, it will return 2017.

Using dates in **financial modeling** is very common and the YEAR function helps extract a year number from a date into a cell. The function can also be used to extract and feed a year value into another formula such as the DATE function.

**Formula****=YEAR(serial\_number)**

The **serial\_number** argument is required. It is the date of the year that we wish to find. Dates should be entered either by using the DATE function or as results of other formulas or functions. Problems can occur if dates are entered as text.

**How to use the YEAR Function in Excel?**

It is a built-in function that can be used as a worksheet function in Excel. To understand the uses of the YEAR function, let's consider a few examples:

### Example 1

Let's assume we shared an Excel file with several users. We want to ensure that a user enters only dates in a certain year. In such a scenario, we can use data validation with a custom formula based on the YEAR function.

Suppose we are given the following data:

|   | A | B                       | C                         | D |
|---|---|-------------------------|---------------------------|---|
| 1 |   |                         |                           |   |
| 2 |   | <b>YEAR Function</b>    |                           |   |
| 3 |   |                         |                           |   |
| 4 |   | <b>Checklist</b>        |                           |   |
| 5 |   |                         |                           |   |
| 6 |   | <b>Task</b>             | <b>Date of Completion</b> |   |
| 7 |   | Sending the RFP         |                           |   |
| 8 |   | Shortlisting Candidates |                           |   |

Using the data below, we want that user needs to enter only dates in B4 and B5. For this, we will set a data validation rule as follows:

Data validation rules will be triggered when any user tries to add or change the cell value.

When the years match, the expression returns TRUE and the validation succeeds. If the years don't match, or if the YEAR function is not able to extract a year, the validation will fail.

The custom validation formula will simply check the year of any date against a hard-coded year value using the YEAR function and return an error if the year is not entered. If we want that a user enters only a date in the current year then the formula to be used is **=YEAR(C5)=YEAR(TODAY())**.

The TODAY function will return the current date on an ongoing basis, so the formula returns TRUE only when a date is in the current year.

## Example 2

If we want, we can extract the year from given set of dates. Suppose we wish to extract Year from the dates below:

|   | A | B                     | C               |
|---|---|-----------------------|-----------------|
| 1 |   |                       |                 |
| 2 |   | <b>YEAR Function</b>  |                 |
| 3 |   |                       |                 |
| 4 |   |                       |                 |
| 5 |   | Date of Inception     | January 1, 2017 |
| 6 |   | Date of Incorporation | 02/07/17        |

Using the YEAR formula, we can extract the year. The formula to use is:

|   | A | B                     | C               | D         | E |
|---|---|-----------------------|-----------------|-----------|---|
| 1 |   |                       |                 |           |   |
| 2 |   | <b>YEAR Function</b>  |                 |           |   |
| 3 |   |                       |                 |           |   |
| 4 |   |                       |                 | Year      |   |
| 5 |   | Date of Inception     | January 1, 2017 | =YEAR(C5) |   |
| 6 |   | Date of Incorporation | 02/07/17        | 2017      |   |

We will get the result below:

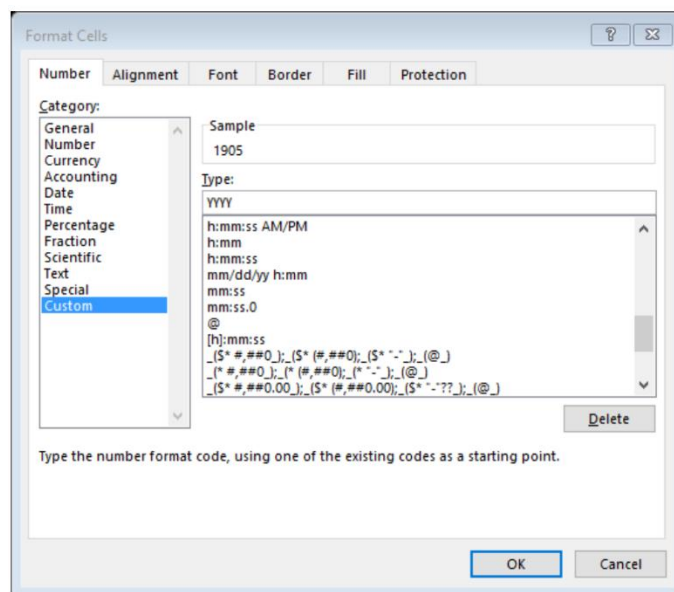
|   | A | B                     | C               | D    | E |
|---|---|-----------------------|-----------------|------|---|
| 1 |   |                       |                 |      |   |
| 2 |   | <b>YEAR Function</b>  |                 |      |   |
| 3 |   |                       |                 |      |   |
| 4 |   |                       |                 | Year |   |
| 5 |   | Date of Inception     | January 1, 2017 | 2017 |   |
| 6 |   | Date of Incorporation | 02/07/17        | 2017 |   |

## Things to remember about the YEAR Function

1. Dates should be supplied to Excel functions as either:
  - Serial numbers; or
  - Reference to cells containing dates; or
  - Date values returned from other Excel formulas.

If we try to input dates as text, then there is a risk that Excel may misinterpret them, due to the different date systems or date interpretation settings on your computer.

2. The value returned by the YEAR function will be Gregorian values regardless of the display format for the supplied date value.
3. The YEAR function perfectly understands dates in all possible formats.
4. If we wish to see only year for dates stored, we can format the cells accordingly. In this case, no formula is needed. We can just open the Format Cells dialog by pressing Ctrl + 1. After that select the Custom category on the Number tab, and enter one of the codes below in the Type box:
  - yy – to display 2-digit years, as 00-99.
  - yyyy – to display 4-digit years, as 1900-9999.



**YEARFRAC** Get the fraction of a year between two dates `start_date` `end_date` `basis`

### What is the YEARFRAC Function?

The YEARFRAC Function is categorized under DATE/TIME functions. YEARFRAC will return the number of days between two dates as a year fraction in Excel.

### Formula

**=YEARFRAC(start\_date, end\_date, [basis])**

The YEARFRAC function uses the following arguments:

1. **Start\_date** (required argument) – It is the start of the period. The function includes the start date in calculations.
2. **End\_date** (required argument) – It is the end of the period. The function includes the end date in calculations.
3. **[basis]** (optional argument) – It specifies the type of day count basis to be used.

Various possible values of [basis] and their meanings are:

| <b>[basis]</b> | <b>Day Count Basis</b> |
|----------------|------------------------|
| 0 or omitted   | US(NASD)30/360         |
| 1              | Actual/Actual          |
| 2              | Actual/360             |
| 3              | Actual/365             |
| 4              | European 30/360        |

### How to use the YEARFRAC Function in Excel?

As a worksheet function, YEARFRAC can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let's consider a few examples:



### Example 1

Let's see a few dates and how the function provides results:

| Start date | End date | Result | Remarks   |
|------------|----------|--------|---|
| 01/30/17   | 07/30/17 | 0.50   | On basis of 1, it gives result as 6 months that is half year.         |
| 01/01/17   | 10/01/17 | 0.75   | On basis of 0, it gives result as 9 months that is three fourth year. |
| 01/01/17   | 12/31/17 | 1.00   | On basis of 0, it gives result as 12 months that is full year.        |

The formula used was:

The screenshot shows the Excel formula bar with the formula `=YEARFRAC(B5,C5,1)`. Below the formula bar, a table titled "YEARFRAC Function" is displayed. The table has four columns: "Start date", "End date", "Result", and "Remarks". The first row of data shows the start date "01/30/17", end date "07/30/17", result "0.50", and remark "On basis of 1 it gives result as 6 months that is half year".

| Start date | End date | Result | Remarks   |
|------------|----------|--------|---|
| 01/30/17   | 07/30/17 | 0.50   | On basis of 1 it gives result as 6 months that is half year |

The results in Excel are shown in the screenshot below:

The screenshot shows the Excel results for the YEARFRAC function. The formula bar displays the text "On basis of 1 it gives result as 6 months that is half year". Below the formula bar, a table titled "YEARFRAC Function" is displayed. The table has four columns: "Start date", "End date", "Result", and "Remarks". The first row of data shows the start date "01/30/17", end date "07/30/17", result "0.50", and remark "On basis of 1 it gives result as 6 months that is half year".

| Start date | End date | Result | Remarks   |
|------------|----------|--------|---|
| 01/30/17   | 07/30/17 | 0.50   | On basis of 1 it gives result as 6 months that is half year         |
| 01/01/17   | 10/01/17 | 0.75   | On basis of 0 it gives result as 9 months that is three fourth year |
| 01/01/17   | 12/31/17 | 1.00   | On basis of 0 it gives result as 12 months that is full year        |

## Example 2

Let's assume we are handling the HR department. We need to calculate a person's age from their birth date. It can be done by using the YEARFRAC, INT, and TODAY functions. In the generic version of the formula above, birthdate is the person's birthday with the year, and TODAY supplies the date on which to calculate age. As the formula uses the TODAY function, it will continue to calculate the correct age in the future as well. Suppose we are given the following data:

|    | A | B                        | C                    |
|----|---|--------------------------|----------------------|
| 1  |   |                          |                      |
| 2  |   | <b>YEARFRAC Function</b> |                      |
| 3  |   |                          |                      |
| 4  |   | <b>Name</b>              | <b>Date of Birth</b> |
| 5  |   | Linda Wilson             | 05/15/89             |
| 6  |   | Tom Cruise               | 03/15/81             |
| 7  |   | Henry Ford               | 05/25/95             |
| 8  |   | Michael Chang            | 10/15/92             |
| 9  |   | Ava Thomas               | 11/15/65             |
| 10 |   | Tracy Brown              | 02/15/85             |

The formula used was:

The screenshot shows the Excel formula bar with the formula `=INT(YEARFRAC(C5,TODAY()))`. Below the formula bar, the spreadsheet view shows the following data:

|   | A | B                        | C                    | D                          | E | F |
|---|---|--------------------------|----------------------|----------------------------|---|---|
| 1 |   |                          |                      |                            |   |   |
| 2 |   | <b>YEARFRAC Function</b> |                      |                            |   |   |
| 3 |   |                          |                      |                            |   |   |
| 4 |   | <b>Name</b>              | <b>Date of Birth</b> | <b>Age</b>                 |   |   |
| 5 |   | Linda Wilson             | 05/15/89             | =INT(YEARFRAC(C5,TODAY())) |   |   |

To work out the fraction of a year as a decimal value, Excel will use whole days between two dates. As all dates are simply serial numbers, the process is straightforward in Excel.

Hence, in this case, we provided birthdate as the start\_date from cell B3, and today's date is supplied as the end\_date by using the TODAY function.

After that, the INT function takes over and rounds down the number to its integer value.

We get the results below:

|    | A | B                        | C                    | D          | E | F |
|----|---|--------------------------|----------------------|------------|---|---|
| 1  |   |                          |                      |            |   |   |
| 2  |   | <b>YEARFRAC Function</b> |                      |            |   |   |
| 3  |   |                          |                      |            |   |   |
| 4  |   | <b>Name</b>              | <b>Date of Birth</b> | <b>Age</b> |   |   |
| 5  |   | Linda Wilson             | 05/15/89             | 28         |   |   |
| 6  |   | Tom Cruise               | 03/15/81             | 36         |   |   |
| 7  |   | Henry Ford               | 05/25/95             | 22         |   |   |
| 8  |   | Michael Chang            | 10/15/92             | 25         |   |   |
| 9  |   | Ava Thomas               | 11/15/65             | 52         |   |   |
| 10 |   | Tracy Brown              | 02/15/85             | 33         |   |   |

If we want to calculate a person's age as of a given date, just replace the TODAY function with that date or a cell reference to that date.

For example, if we are holding a competition for people aged below 35 years old, we can use the following formula:

**=IF(INT(YEARFRAC(A1,TODAY()))<35,"Eligible","Not Eligible")**

If we wish to calculate the age on a specific date, given a birth date, we can use the DATE function instead of the TODAY function. The formula will be:

**=INT(YEARFRAC(A1,DATE(2017,1,1)))**

### Things to remember about the YEARFRAC Function

1. #NUM! error – Occurs when the given basis argument is less than 0 or greater than 4.
2. #VALUE! error – Occurs when:
  1. The start\_date or end\_date arguments are not valid dates.
  2. The given [basis] argument is non-numeric.

## Financial

**DURATION** Get the duration of a security

settlement

maturity

coupon

yield

frequency

basis

### What is the DURATION Function?

The DURATION function is categorized under **Financial functions**. It helps to calculate the Macauley Duration. The function calculates the duration of a security that pays interest on a periodic basis with a par value of \$100.

DURATION is commonly used by **Portfolio Managers** who use the immunization strategy. Apart from it, the function is also useful in financial modeling, particularly in predicting future **cash flows** of investments.

### Formula

**=DURATION(settlement, maturity, coupon, yield, frequency, [basis])**

The DURATION function uses the following arguments:

1. **Settlement** (required argument) – It is the security's settlement date or the date on which the coupon is purchased.
2. **Maturity** (required argument) – It is the security's maturity date or the date on which the coupon expires.
3. **Coupon** (required argument) – It is the security's coupon rate.
4. **Yield** (required argument) – It is the security's annual yield.
5. **Frequency** (required argument) – It is the number of coupon payments per year. For annual payments, the frequency is = 1; for semiannual, frequency is = 2; and for quarterly, frequency = 4.
6. **Basis** (optional argument) – It is the type of day count basis to be used. The possible values of basis are:

| Basis        | Day Count Basis |
|--------------|-----------------|
| 0 or omitted | US(NASD) 30/360 |
| 1            | Actual/actual   |
| 2            | Actual/360      |
| 3            | Actual/365      |
| 4            | European 30/360 |

Remember that the date arguments should be encoded into the DURATION function as either:

- References to cells containing dates; or
- Dates returned from formulas

Suppose If we try to input text representations of dates into Excel functions, they may be interpreted differently, depending on the date system and date interpretation settings on our computer.

### **How to use the DURATION Function in Excel?**

As a worksheet function, DURATION can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let us consider a few examples:

### Example 1

In this example, we will calculate the duration of a coupon purchased on April 1, 2017, with a maturity date of March 31, 2025 and a coupon rate of 6%. The yield is 5% and payments are made quarterly.

| IRR |                          | X                        |   | ✓ |  | fx |  | =DURATION(C4,C5,6%,5%,4) |  |
|-----|--------------------------|--------------------------|---|---|--|----|--|--------------------------|--|
| A   | B                        | C                        | D | E |  |    |  |                          |  |
| 1   |                          |                          |   |   |  |    |  |                          |  |
| 2   | <b>DURATION Function</b> |                          |   |   |  |    |  |                          |  |
| 3   |                          |                          |   |   |  |    |  |                          |  |
| 4   | Settlement date          | April 1, 2017            |   |   |  |    |  |                          |  |
| 5   | Maturity                 | March 31, 2025           |   |   |  |    |  |                          |  |
| 6   |                          |                          |   |   |  |    |  |                          |  |
| 7   |                          | =DURATION(C4,C5,6%,5%,4) |   |   |  |    |  |                          |  |
| 8   |                          |                          |   |   |  |    |  |                          |  |

The function returns a duration of 6.46831 years.

| C7 |                          | X              |   | ✓ |  | fx |  | =DURATION(C4,C5,6%,5%,4) |  |
|----|--------------------------|----------------|---|---|--|----|--|--------------------------|--|
| A  | B                        | C              | D | E |  |    |  |                          |  |
| 1  |                          |                |   |   |  |    |  |                          |  |
| 2  | <b>DURATION Function</b> |                |   |   |  |    |  |                          |  |
| 3  |                          |                |   |   |  |    |  |                          |  |
| 4  | Settlement date          | April 1, 2017  |   |   |  |    |  |                          |  |
| 5  | Maturity                 | March 31, 2025 |   |   |  |    |  |                          |  |
| 6  |                          |                |   |   |  |    |  |                          |  |
| 7  |                          | 6.46831        |   |   |  |    |  |                          |  |

As we omitted the basis argument, the DURATION function took the days count as US(NASD) 30/360. As it uses Macaulay Duration, the formula used is:

$$\frac{\sum_{t=1}^n \frac{t \cdot C}{(1+y)^t} + \frac{n \cdot M}{(1+y)^n}}{\text{Current Bond Price}}$$

It calculates the weighted average term to maturity of the cash flows from a bond. The weight of each cash flow is determined by dividing the present value of the cash flow by the bond price.

**Things to remember about the DURATION Function:**

1. #NUM! error – Occurs if either:
  1. The supplied settlement date is  $\geq$  maturity date; or
  2. Invalid numbers are supplied for the coupon, yld, frequency or [basis] arguments, i.e. if either: coupon  $< 0$ ; yld  $< 0$ ; frequency is not equal to 1, 2 or 4; or [basis] is supplied and is not equal to 0, 1, 2, 3 or 4).
3. #VALUE! error – Occurs if either:
  1. Any of the given arguments is non-numeric; or
  2. One or both of the given settlement or maturity dates are not valid Excel dates.
4. Settlement, maturity, frequency, and basis are truncated to integers.
5. In MS Excel, dates are stored as sequential serial numbers so they can be used in calculations. By default, January 1, 1900 is serial number 1, and January 18, 2018 is serial number 43118, because it is 43,118 days after January 1, 1900.

**NPV** Get the Net Present Value (NPV) for periodic cash flows

rate

value1

value2

### What is the NPV Function?

The NPV Function is categorized under **Financial functions**. It will calculate the **Net Present Value (NPV)** for periodic cash flows. The NPV will be calculated for an investment by using a discount rate and series of future payments and income.

In **financial modeling**, the NPV function is useful in determining the value of an investment or understanding the feasibility of a project. **It should be noted that it's better for analysts to use the XNPV function instead of the regular NPV function.**

### Formula

**=NPV(rate,value1,[value2],...)**

The NPV function uses the following arguments:

1. **Rate** (required argument) – It is the rate of discount over the length of the period.
2. **Value1, Value2** – Value1 is required option. They are numeric values that represent series of payments and income where:
  - Negative payments represent outgoing payments.
  - Positive payments represent incoming payments.

The NPV function uses the following equation to calculate the Net Present Value of an Investment:

$$NPV = \sum_{i=1}^n \frac{Values_i}{(1 + rate)^i}$$

### How to use the NPV Function in Excel?

To understand the uses of the function, let's consider a few examples:



### Example 1

Suppose we are given the following data on cash inflows and outflows:

|    | A | B                   | C               | D |
|----|---|---------------------|-----------------|---|
| 1  |   |                     |                 |   |
| 2  |   | <b>NPV Function</b> |                 |   |
| 3  |   |                     |                 |   |
| 4  |   |                     |                 |   |
| 5  |   | <b>Period</b>       | <b>Cashflow</b> |   |
| 6  |   | 0                   | (800)           |   |
| 7  |   | 1                   | 100             |   |
| 8  |   | 2                   | 200             |   |
| 9  |   | 3                   | 300             |   |
| 10 |   | 4                   | 400             |   |
| 11 |   | 5                   | 500             |   |
| 12 |   |                     |                 |   |
| 13 |   | Required return     | 10%             |   |

The required rate of return is 10%. To calculate the NPV, we will use the formula below:

| IRR |   | =NPV(C13,C6:C11)    |                  |
|-----|---|---------------------|------------------|
| A   | B | C                   | D                |
| 1   |   |                     |                  |
| 2   |   | <b>NPV Function</b> |                  |
| 3   |   |                     |                  |
| 4   |   |                     |                  |
| 5   |   | <b>Period</b>       | <b>Cashflow</b>  |
| 6   |   | 0                   | (800)            |
| 7   |   | 1                   | 100              |
| 8   |   | 2                   | 200              |
| 9   |   | 3                   | 300              |
| 10  |   | 4                   | 400              |
| 11  |   | 5                   | 500              |
| 12  |   |                     |                  |
| 13  |   | Required return     | 10%              |
| 14  |   |                     |                  |
| 15  |   | NPV                 | =NPV(C13,C6:C11) |

We get the result below:

| Period | Cashflow |
|--------|----------|
| 0      | (800)    |
| 1      | 100      |
| 2      | 200      |
| 3      | 300      |
| 4      | 400      |
| 5      | 500      |

Required return 10%

NPV \$241.14

The **NPV formula** is based on future cash flows. If the first cash flow occurs at the start of the first period, the first value must be added to the NPV result, not included in the values arguments.

### Things to remember about the NPV Function

1. Arguments that are empty cells, logical values, or text representations of numbers, error values, or text that cannot be translated into numbers are ignored.
2. If an argument is an array or reference, only numbers in that array or reference are counted. Empty cells, logical values, text, or error values in the array or reference are ignored.
3. We need to enter our payments and income values in the right sequence as NPV uses the order of value1, value2,... to interpret the order of cash flows.
4. Value1, value2, ... must be equally spaced in time and occur at the end of each period.
5. The NPV function is related to the IRR function (Internal Rate of Return). IRR is the rate for which the NPV equals zero.
6. The primary difference between PV function and NPV is that PV allows cash flows to begin either at the end or at the beginning of the period.

**PMT** Get the periodic payment for a loan **rate** **nper** **pv** **fv** **type**

## What is the PMT Function?

The PMT function is categorized under financial **Excel functions**. The function helps calculate the total payment (principal and interest) required to settle a loan or an investment with a fixed interest rate over a specific time period.

## Formula

**=PMT(rate, nper, pv, [fv], [type])**

The PMT function uses the following arguments:

1. **Rate** (required argument) – The interest rate of the loan.
2. **Nper** (required argument) – The total number of payments for the loan taken.
3. **Pv** (required argument) – The present value or total amount that a series of future payments is worth now. It is also termed as principal.
4. **Fv** (optional argument) – The FV or a cash balance we want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0.
5. **Type** (optional argument) – The security's price. It is the type of day count basis to use. The possible values of basis are:

| Basis        | Day count basis |
|--------------|-----------------|
| 0 or omitted | US(NASD) 30/360 |
| 1            | Actual/actual   |
| 2            | Actual/360      |
| 3            | Actual/365      |
| 4            | European 30/360 |

## How to use the PMT Function in Excel?

As a worksheet function, the PMT function can be entered as part of a formula in a cell of a worksheet. To understand the uses of PMT, let us consider an example:

### Example 1

Let's assume that we need to invest in such a manner that after two years, we'll receive \$75,000. The rate of interest is 3.5% per year and the payment will be made at the start of each month. The details are:

|   | A | B                   | C      |
|---|---|---------------------|--------|
| 1 |   |                     |        |
| 2 |   | <b>PMT Function</b> |        |
| 3 |   |                     |        |
| 4 |   | Rate of Interest    | 3.50%  |
| 5 |   | nper                | 24     |
| 6 |   | PV                  | 75,000 |
| 7 |   | FV                  | -      |
| 8 |   | type                | -      |

The formula used is:

|    | A | B                   | C                 | D | E |
|----|---|---------------------|-------------------|---|---|
| 1  |   |                     |                   |   |   |
| 2  |   | <b>PMT Function</b> |                   |   |   |
| 3  |   |                     |                   |   |   |
| 4  |   | Rate of Interest    | 3.50%             |   |   |
| 5  |   | nper                | 24                |   |   |
| 6  |   | PV                  | 75,000            |   |   |
| 7  |   | FV                  | -                 |   |   |
| 8  |   | type                | -                 |   |   |
| 9  |   |                     |                   |   |   |
| 10 |   |                     |                   |   |   |
| 11 |   | Monthly investments | =PMT(C4/12,C5,C6) |   |   |

We get the results below:

|    | A | B                   | C            | D |
|----|---|---------------------|--------------|---|
| 1  |   |                     |              |   |
| 2  |   | <b>PMT Function</b> |              |   |
| 3  |   |                     |              |   |
| 4  |   | Rate of Interest    | 3.50%        |   |
| 5  |   | nper                | 24           |   |
| 6  |   | PV                  | 75,000       |   |
| 7  |   | FV                  | -            |   |
| 8  |   | type                | -            |   |
| 9  |   |                     |              |   |
| 10 |   |                     |              |   |
| 11 |   | Monthly investments | (\$3,240.20) |   |

The above function returns PMT as \$3,240.20. It is the monthly cash outflow required to realize \$75,000 in two years. In this example:

- The payments into the investment are on a monthly basis. Hence, the annual interest rate is converted to a monthly rate. Also, we converted the years into months:  $2 \times 12 = 24$ .
- The [type] argument is set to 1 to indicate that the payment of the investment will be made at the beginning of each quarter.
- As per the general cash flow convention, outgoing payments are represented by negative numbers and incoming payments are represented by positive numbers.
- As the value returned is negative, it indicates an outgoing payment is to be made.
- The value 3,240.20 includes the principal and interest but no taxes, reserve payments, or fees that are sometimes associated with loans.

#### **Few things to remember about the PMT Function:**

1. #NUM! error – Occurs when:
  1. The given rate value is less than or equal to -1.
  2. The given nper value is equal to 0.
3. #VALUE! error – Occurs when any of the arguments provided are non-numeric.
4. When calculating monthly or quarterly payments, we need to convert annual interest rates or the number of periods to months or quarters.
5. If we wish to find out the total amount that was paid for the duration of the loan, we need to multiply the PMT as calculated by nper.

**PPMT** Get principal for given period

rate

nper

pv

fv

type

### What is the PPMT Function?

The PPMT Function is categorized under **Financial functions**. The function will calculate the payment on the principal for an investment based on periodic, constant payments and a fixed interest rate for a given period of time.

In **financial analysis**, the PPMT function is useful in understanding the principal components of the total payments made for the loan taken.

### Formula

**=PPMT( rate, per, nper, pv, [fv], [type] )**

The PPMT function uses the following arguments:

1. **Rate** (required argument) – It is the interest rate per period.
2. **Per** (required argument) – It is the bond's maturity date, that is, the date when bond expires.
3. **Nper** (required argument) – It is the total number of payment periods in an annuity.
4. **Pv** (required argument) – It is the present value of the loan/investment. It is the total amount that a series of future payments is worth now.
5. **Fv** (optional argument) – It specifies the future value of the loan/investment at the end of nper payments. If omitted, [fv] takes on the default value of 0.
6. **Type** (optional argument) – It specifies whether the payment is made at the start or the end of the period. It can assume a value of 0 or 1. If it is 0, it means the payment is made at the end of the period; and if 1, the payment is made at the start. If we omit the [type] argument, it will take on the default value of 0, denoting payments made at the end of the period.

### How to use the PPMT Function in Excel?

As a worksheet function, PPMT can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let us consider an example:

### Example 1

We need to calculate the payment on the principal for months 1 and 2 on a \$50,000 loan, which is to be paid off in full after 5 years. Interest is charged at a rate of 5% per year and the loan repayments are to be made at the end of each month. The formula used provides a reference to the relevant cells.

|    |                      | IRR                   |                       |        |
|----|----------------------|-----------------------|-----------------------|--------|
|    |                      | =PPMT(C5/12,C7,C6,C4) |                       |        |
|    | A                    | B                     | C                     | D      |
| 1  |                      |                       |                       |        |
| 2  |                      | <b>PPMT Function</b>  |                       |        |
| 3  |                      |                       |                       |        |
| 4  | Loan amount          |                       | 50,000                | 50,000 |
| 5  | Interest rate        |                       | 5%                    | 5%     |
| 6  | Term                 |                       | 60                    | 60     |
| 7  | Per                  |                       | 1.00                  | 2.00   |
| 8  |                      |                       |                       |        |
| 9  |                      |                       |                       |        |
| 10 | Payment on principal |                       | =PPMT(C5/12,C7,C6,C4) |        |

We get the results below:

|    |                      | C10                   |            |            |
|----|----------------------|-----------------------|------------|------------|
|    |                      | =PPMT(C5/12,C7,C6,C4) |            |            |
|    | A                    | B                     | C          | D          |
| 1  |                      |                       |            |            |
| 2  |                      | <b>PPMT Function</b>  |            |            |
| 3  |                      |                       |            |            |
| 4  | Loan amount          |                       | 50,000     | 50,000     |
| 5  | Interest rate        |                       | 5%         | 5%         |
| 6  | Term                 |                       | 60         | 60         |
| 7  | Per                  |                       | 1.00       | 2.00       |
| 8  |                      |                       |            |            |
| 9  |                      |                       |            |            |
| 10 | Payment on principal |                       | (\$735.23) | (\$738.29) |



For month 2, we used the formula below:

|    | A | B                    | C          | D                     | E | F |
|----|---|----------------------|------------|-----------------------|---|---|
| 1  |   |                      |            |                       |   |   |
| 2  |   | <b>PPMT Function</b> |            |                       |   |   |
| 3  |   |                      |            |                       |   |   |
| 4  |   | Loan amount          | 50,000     | 50,000                |   |   |
| 5  |   | Interest rate        | 5%         | 5%                    |   |   |
| 6  |   | Term                 | 60         | 60                    |   |   |
| 7  |   | Per                  | 1.00       | 2.00                  |   |   |
| 8  |   |                      |            |                       |   |   |
| 9  |   |                      |            |                       |   |   |
| 10 |   | Payment on principal | (\$735.23) | =PPMT(D5/12,D7,D6,D4) |   |   |

We get the results below:

|    | A | B                    | C          | D          | E |
|----|---|----------------------|------------|------------|---|
| 1  |   |                      |            |            |   |
| 2  |   | <b>PPMT Function</b> |            |            |   |
| 3  |   |                      |            |            |   |
| 4  |   | Loan amount          | 50,000     | 50,000     |   |
| 5  |   | Interest rate        | 5%         | 5%         |   |
| 6  |   | Term                 | 60         | 60         |   |
| 7  |   | Per                  | 1.00       | 2.00       |   |
| 8  |   |                      |            |            |   |
| 9  |   |                      |            |            |   |
| 10 |   | Payment on principal | (\$735.23) | (\$738.29) |   |

The above PPMT function returns the value \$735.23 (rounded off to 2 decimal points). In the above example:

1. We made monthly payments, so it is necessary to convert the annual interest rate of 5% into the monthly rate ( $=5\%/12$ ), and the number of periods from years into months ( $=5*12$ ).
2. Since the forecast value is zero and we need to make monthly payments, we omitted the FV and type arguments.
3. We get a negative value, which signifies outgoing payments.

**Few things to remember about the PPMT Function:**

1. #NUM! error – Occurs if the given per argument is less than 0 or is greater than the supplied value of nper.
2. VALUE! error – Occurs if any of the given arguments are non-numeric.
3. An error can arise if we forget to convert the interest rate or the number of periods to months or quarters when calculating monthly or quarterly payments. For the monthly rate, we need to divide annual rate by 12 and by 4 for the quarterly rate.

**XIRR**

Get the Internal Rate of Return (IRR) for a series of cash flows that may not be periodic

value

dates

guess

### What is the XIRR Function?

The XIRR function is categorized under financial functions and will calculate the **Internal Rate of Return (IRR)** for a series of cash flows that may not be periodic by assigning specific dates to each individual cash flow. The main benefit of using the XIRR Excel function is that these unevenly timed cash flows can be accurately modeled. To learn more, read [why to always use XIRR over IRR](#) in Excel modeling.

In **financial modeling**, the XIRR function is useful in determining the value of an investment or understanding the feasibility of a project without periodic cash flows. It helps us understand the rate of return earned on an investment. Hence, it is commonly used in finance, particularly when choosing between investments.

### Formula

**=XIRR(values, dates,[guess])**

The formula uses the following arguments:

1. **Values** (required argument) – It is the array of values that represent the series of cash flows. Instead of an array, it can be a reference to a range of cells containing values.
2. **Dates** (required argument) – It is a series of dates that correspond to the given values. Subsequent dates should be later than the first date as the first date is the start date and subsequent dates are future dates of outgoing payments or income.
3. **[guess]** (optional argument) – It is an initial guess of what the IRR would be. If omitted, Excel takes the default value that is 10%.

Excel uses an iterative technique for calculating XIRR. Using a changing rate (starting with [guess]), XIRR cycles through the calculation until the result is accurate within 0.000001%.

### How to use the XIRR Function in Excel?

To understand the uses of the XIRR function, let's consider a few examples:

## XIRR Example

Suppose a project started on January 1, 2018. The project gives us cash flows in the middle of the first year, after 6 months, then at end of 1.5 years, 2nd year, 3.5th year and annually thereafter. The data given is shown below:

|    | A | B                    | C               | D |
|----|---|----------------------|-----------------|---|
| 1  |   |                      |                 |   |
| 2  |   | <b>XIRR Function</b> |                 |   |
| 3  |   |                      |                 |   |
| 4  |   | <b>Date</b>          | <b>Cashflow</b> |   |
| 5  |   | 08/06/2018           | (800)           |   |
| 6  |   | 08/08/2019           | 100             |   |
| 7  |   | 08/12/2020           | 200             |   |
| 8  |   | 08/06/2021           | 300             |   |
| 9  |   | 08/06/2022           | 400             |   |
| 10 |   | 08/01/2023           | 500             |   |
| 11 |   |                      |                 |   |

The formula to use will be:

|    | A | B                    | C               | D | E           | F                    |
|----|---|----------------------|-----------------|---|-------------|----------------------|
| 1  |   |                      |                 |   |             |                      |
| 2  |   | <b>XIRR Function</b> |                 |   |             |                      |
| 3  |   |                      |                 |   |             |                      |
| 4  |   | <b>Date</b>          | <b>Cashflow</b> |   |             |                      |
| 5  |   | 08/06/2018           | (800)           |   |             |                      |
| 6  |   | 08/08/2019           | 100             |   |             |                      |
| 7  |   | 08/12/2020           | 200             |   | <b>XIRR</b> | =XIRR(C5:C10,B5:B10) |
| 8  |   | 08/06/2021           | 300             |   |             |                      |
| 9  |   | 08/06/2022           | 400             |   |             |                      |
| 10 |   | 08/01/2023           | 500             |   |             |                      |
| 11 |   |                      |                 |   |             |                      |

We will leave the guess as blank so Excel takes the default value of 10%.

We get the result below:

|    | A | B                    | C               | D | E           | F   |
|----|---|----------------------|-----------------|---|-------------|-----|
| 1  |   |                      |                 |   |             |     |
| 2  |   | <b>XIRR Function</b> |                 |   |             |     |
| 3  |   |                      |                 |   |             |     |
| 4  |   | <b>Date</b>          | <b>Cashflow</b> |   |             |     |
| 5  |   | 08/06/2018           | (800)           |   |             |     |
| 6  |   | 08/08/2019           | 100             |   |             |     |
| 7  |   | 08/12/2020           | 200             |   | <b>XIRR</b> | 20% |
| 8  |   | 08/06/2021           | 300             |   |             |     |
| 9  |   | 08/06/2022           | 400             |   |             |     |
| 10 |   | 08/01/2023           | 500             |   |             |     |
| 11 |   |                      |                 |   |             |     |

### Things to remember about the XIRR Function

1. Numbers in dates are truncated to integers.
2. XNPV and XIRR are closely related. The rate of return calculated by XIRR is the interest rate corresponding to  $XNPV = 0$ .
3. Dates should be entered as references to cells containing dates or values returned from Excel formulas.
4. #NUM! error – Occurs if either:
  1. The values and dates arrays are of different lengths;
  2. The given arrays do not contain at least one negative and at least one positive value;
  3. Any of the given dates precedes the first date provided; or
  4. The calculation fails to converge after 100 iterations.
5. #VALUE! error – Occurs when either of the dates given cannot be recognized by Excel as valid dates.

**XNPV** Get the Net Present Value (NPV) for a series of cash flows that may not be periodic

rate

values

dates

### Why use the XNPV Function in Excel?

The XNPV function in Excel uses specific dates that correspond to each cash flow being discounted in the series, whereas the regular NPV function automatically assumes all the time periods are equal. For this reason, **the XNPV function is far more precise** and should be used instead of the regular **NPV function**.

More learning: [read CFI's list of top Excel formulas](#).

## XNPV Function

The diagram shows an Excel spreadsheet with columns A through I and rows 1 through 14. The following table summarizes the data layout:

|    | A      | B | C               | D | E | F | G | H | I |
|----|--------|---|-----------------|---|---|---|---|---|---|
| 1  |        |   |                 |   |   |   |   |   |   |
| 2  |        |   |                 |   |   |   |   |   |   |
| 3  |        |   |                 |   |   |   |   |   |   |
| 4  | Dates  |   | [Dark Blue Bar] |   |   |   |   |   |   |
| 5  |        |   |                 |   |   |   |   |   |   |
| 6  |        |   |                 |   |   |   |   |   |   |
| 7  | Values |   | [Teal Bar]      |   |   |   |   |   |   |
| 8  |        |   |                 |   |   |   |   |   |   |
| 9  |        |   |                 |   |   |   |   |   |   |
| 10 |        |   |                 |   |   |   |   |   |   |
| 11 | Rate   |   | [Orange Bar]    |   |   |   |   |   |   |
| 12 |        |   |                 |   |   |   |   |   |   |
| 13 |        |   |                 |   |   |   |   |   |   |
| 14 |        |   |                 |   |   |   |   |   |   |

### What is the XNPV formula?

The XNPV formula in Excel requires the user to select a discount rate, a series of cash flows, and a series of corresponding dates for each cash flow.

The Excel formula for XNPV is:

**=XNPV(Rate, Cash Flows, Dates of Cash Flow)**

The XNPV function uses the following three components:

1. **Rate** – The discount rate to be used over the length of the period (see hurdle rate and WACC articles to learn about what rate to use).
2. **Values** (Cash Flows) – This is an array of numeric values that represent the payments and income where:
  - Negative values are treated as outgoing payments (negative cash flow).
  - Positive values are treated as income (positive cash flow).
3. **Dates** (of Cash Flows) – It is an array of dates corresponding to an array of payments. The date array should be of the same length as the values array.

The function uses the following equation to calculate the Net Present Value of an investment:

$$XNPV = \sum_{i=1}^N \frac{P_i}{(1 + rate)^{\frac{d_i - d_1}{365}}}$$

Where:

$d_i$  = the  $i$ 'th payment date

$d_1$  = the 0'th payment date

$P_i$  = the  $i$ 'th payment

Please see the example below for a detailed breakdown of how to use XNPV in Excel.

### Example of XNPV in Excel

Below is a screenshot of an Example of the function being used in Excel to calculate the **Net Present Value** of a series of cash flows based on specific dates.

Key assumptions in the XNPV example:

- The discount rate is 10%
- The start date is June 30, 2018 (date we are discounting the cash flows back to)
- Cash flows are received on the exact date they correspond to
- The time between the start date and the first cash flow is only 6 months

|    | A | B                      | C                     | D          | E          | F          | G          | H          | I          | J          | K          | L          | M          |
|----|---|------------------------|-----------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| 1  |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 2  |   | Discount Rate          | 10%                   |            |            |            |            |            |            |            |            |            |            |
| 3  |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 4  |   | <b>Period</b>          | <b>Start</b>          | <b>1</b>   | <b>2</b>   | <b>3</b>   | <b>4</b>   | <b>5</b>   | <b>6</b>   | <b>7</b>   | <b>8</b>   | <b>9</b>   | <b>10</b>  |
| 5  |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 6  |   | Dates                  | 6/30/2018             | 12/31/2018 | 12/31/2019 | 12/31/2020 | 12/31/2021 | 12/31/2022 | 12/31/2023 | 12/31/2024 | 12/31/2025 | 12/31/2026 | 12/31/2027 |
| 7  |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 8  |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 9  |   | Undiscounted Cash Flow | 0                     | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  | 120,000.0  |
| 10 |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 11 |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 12 |   | XNPV                   | =XNPV(C2,C9:M9,C6:M6) |            | 772,830.7  |            |            |            |            |            |            |            |            |
| 13 |   |                        |                       |            |            |            |            |            |            |            |            |            |            |
| 14 |   | Regular NPV            |                       |            | 670,316.4  |            |            |            |            |            |            |            |            |
| 15 |   |                        |                       |            |            |            |            |            |            |            |            |            |            |

Based on the above, the XNPV formula produces a value of \$772,830.7 while the regular NPV formula produces a value of \$670,316.4.

The reason for this difference is that XNPV recognizes that time period between the start date and first cash flow is only 6 months, while the NPV function treats it as a full-time period.



### **Download the XNPV template**

If you'd like to incorporate this function in your own financial modeling and valuation work, please feel free to [Download CFI's XNPV function template](#) and use it as you see it.

It could be a good idea to experiment with changing the dates around and seeing the impact on valuation, or the relative difference between XNPV vs NPV in the Excel model.

### **XNPV vs NPV implications**

The results of the comparisons of XNPV vs NPV formulas produces an interesting result and some very important implications for a financial analyst. Imagine if the analyst were valuing a security and didn't use the proper time periods that XNPV takes into account... they would be undervaluing the security by a meaningful amount!

## Use in financial modeling

XNPV is used routinely in **financial modeling** as a means of calculating the net present value (NPV) of an investment opportunity.

For all types of financial models, XNPV and XIRR are highly recommended over their date-less counterparts NPV and IRR.

While the added precision will have you feeling more confident about your analysis, the only downside is that you have to pay careful attention to the dates in your spreadsheet and make sure the start date always reflects what it should.

For a transaction, like with a **Leveraged Buyout (LBO)** or an acquisition, it's important to be precise about the closing date of the deal. For example, you may be building the model now, but the closing date will likely be several months in the future.

|     | A   | B | C    | D | E                  | F                  | G                | H              | I                | J                | K                | L |
|-----|---|---|------|---|--------------------|--------------------|------------------|----------------|------------------|------------------|------------------|---|
| 1   | © Corporate Finance Institute. All rights reserved. |   |      |   | Startup year       |                    |                  |                |                  |                  | Terminal year    |   |
| 2   |   |   |      |   | 2016               | 2017               | 2018             | 2019           | 2020             | 2021             | 2022             |   |
| 3   | Balance Sheet Check                                 |   |      |   | OK                 | OK                 | OK               | OK             | OK               | OK               | OK               |   |
| 266 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 267 | <b>Valuation</b>                                    |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 268 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 269 | EBT   |   |      |   | (2,573,040)        | (1,340,320)        | 57,140           | 1,233,760      | 2,782,080        | 4,941,600        | 5,738,560        |   |
| 270 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 271 | <b>Net Operating Loss</b>                           |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 272 | Opening Balance                                     |   |      |   | 0                  | 2,573,040          | 3,913,360        | 3,856,220      | 2,622,460        | 0                | 0                |   |
| 273 | Current Loss  |   |      |   | 2,573,040          | 1,340,320          | 0                | 0              | 0                | 0                | 0                |   |
| 274 | Sub Total   |   |      |   | 2,573,040          | 3,913,360          | 3,913,360        | 3,856,220      | 2,622,460        | 0                | 0                |   |
| 275 | Loss Used   |   |      |   | 0                  | 0                  | 57,140           | 1,233,760      | 2,622,460        | 0                | 0                |   |
| 276 | Closing Balance                                     |   |      |   | 2,573,040          | 3,913,360          | 3,856,220        | 2,622,460      | 0                | 0                | 0                |   |
| 277 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 278 | <b>Free Cash Flow</b>                               |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 279 | Earning Before Tax                                  |   |      |   | (2,573,040)        | (1,340,320)        | 57,140           | 1,233,760      | 2,782,080        | 4,941,600        | 5,738,560        |   |
| 280 | Interest  |   |      |   | 0                  | 0                  | 0                | 0              | 0                | 0                | 0                |   |
| 281 | EBIT  |   |      |   | (2,573,040)        | (1,340,320)        | 57,140           | 1,233,760      | 2,782,080        | 4,941,600        | 5,738,560        |   |
| 282 | Less Taxes  |   |      |   | 0                  | 0                  | 0                | 0              | 695,520          | 1,235,400        | 1,434,640        |   |
| 283 | Less Capex  |   |      |   | 500,000            | 100,000            | 100,000          | 550,000        | 250,000          | 1,100,000        | 600,000          |   |
| 284 | Plus Depreciation                                   |   |      |   | 100,000            | 120,000            | 140,000          | 250,000        | 300,000          | 420,000          | 520,000          |   |
| 285 | Less Changes in Working Capital                     |   |      |   | 72,314             | 289,534            | 434,654          | 732,196        | 573,939          | 806,624          | 567,112          |   |
| 286 | <b>Unlevered Free Cash Flow</b>                     |   |      |   | <b>(3,045,354)</b> | <b>(1,609,854)</b> | <b>(337,514)</b> | <b>201,564</b> | <b>1,562,621</b> | <b>2,219,576</b> | <b>3,656,808</b> |   |
| 287 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 288 | <b>DCF Valuation</b>                                |   |      |   |                    |                    |                  |                |                  |                  |                  |   |
| 289 | NPV of Forecast                                     |   | 20%  |   | (1,362,003)        |                    |                  |                |                  |                  |                  |   |
| 290 | Terminal Value (EBITDA multiple)                    |   | 8.0x |   | 50,068,480         |                    |                  |                |                  |                  |                  |   |
| 291 | NPV of Terminal Value                               |   |      |   | 13,973,194         |                    |                  |                |                  |                  |                  |   |
| 292 | <b>Total Enterprise Value</b>                       |   |      |   | <b>12,611,191</b>  |                    |                  |                |                  |                  |                  |   |
| 293 |   |   |      |   |                    |                    |                  |                |                  |                  |                  |   |

To see the function in action, check out CFI's **financial modeling courses!**

### Things to remember about the XNPV function

Below is a helpful list of points to remember:

1. Numbers in dates are truncated to integers.
2. XNPV doesn't discount the initial cash flow (it brings all cash flows back to the date of the first cash flow). Subsequent payments are discounted based on a 365-day year.
3. #NUM! error – Occurs when either:
  - The values and dates arrays are of different lengths; or
  - Any of the other dates are earlier than the start date.
4. #VALUE! error – Occurs when either:
  - The values or rates arguments are non-numeric; or
  - The given dates are not recognized by Excel as valid dates.

### XIRR vs IRR

To learn more about [XIRR vs IRR](#) we have created a similar guide, which we highly recommend you check out to further solidify the concept. For the same reasons as noted above in this guide, it's equally important to use specific dates when calculating the [internal rate of return](#) on an investment.

If you're considering a career in investment banking or private equity you will be required to use these functions extensively.

**YIELD** Get the yield on a security

settlement

maturity

rate

pr

redemption

frequency

basis

## What is the YIELD Function?

The YIELD Function is categorized under **Financial functions**. It will calculate the yield on a security that pays periodic interest. The function is generally used to calculate the bond yield.

As a **financial analyst**, we often calculate the yield on a bond to determine the income that would be generated in a year. Yield is different from the rate of return as the latter return is the gain already earned while former is the prospective return.

## Formula

= YIELD(settlement, maturity, rate, pr, redemption, frequency, [basis])

This function uses the following arguments:

1. **Settlement** (required argument) – It is the settlement date of the security. It is a date after the security is traded to the buyer that is after the issue date.
2. **Maturity** (required argument) – It is the maturity date of the security. It is the date when the security expires.
3. **Rate** (required argument) – It is the annual coupon rate.
4. **Pr** (required argument) – It is the price of security per \$100 face value.
5. **Redemption** (required argument) – It is the redemption value per \$100 face value.
6. **Frequency** (required argument) – It is the number of coupon payments per year. It must be either of the following: 1 – Annually, 2 – Semi-annually, 4 – Quarterly
7. **[basis]** (optional argument) – It specifies the financial day count basis that is used by security. The possible values are:

| Basis        | Day Count basis |
|--------------|-----------------|
| 0 or omitted | US(NASD) 30/360 |
| 1            | Actual/actual   |
| 2            | Actual/360      |
| 3            | Actual/365      |
| 4            | European 30/360 |

The settlement and maturity dates should be supplied to the YIELD function as either:

- References to cells containing dates; or
- Dates returned from formulas.

## How to use the YIELD Function in Excel?

As a worksheet function, YIELD can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, and let us consider an example:

### Example

Suppose we are given the following data:

- Settlement date: **01/01/2017**
- Maturity date: **6/30/2019**
- Rate of interest: **10%**
- Price per \$100 FV: **\$101**
- Redemption value: **\$100**
- Payment terms: **Quarterly**

We can use the function to find out the yield. The formula to use will be:

| YIELD Function     |           |
|--------------------|-----------|
| Settlement date    | 01/01/17  |
| Maturity date      | 06/30/19  |
| Rate of Interest   | 10%       |
| Price per \$100 FV | \$101     |
| Redemption Value   | \$100     |
| Payment terms      | Quarterly |

Yield      =YIELD(C4,C5,C6,C7,C8,4,0)

We get the result below:

| YIELD Function     |           |
|--------------------|-----------|
| Settlement date    | 01/01/17  |
| Maturity date      | 06/30/19  |
| Rate of Interest   | 10%       |
| Price per \$100 FV | \$101     |
| Redemption Value   | \$100     |
| Payment terms      | Quarterly |
| Yield              | 9.55%     |

In the above example:

1. We used as basis the US (NASD) 30/360 day basis.
2. As recommended by Microsoft, the date arguments were entered as references to cells containing dates.

**Few things to remember about the YIELD Function:**

1. #NUM! error – Occurs when:
  - The settlement date provided is greater than or equal to the maturity date.
  - We provided invalid numbers for the rate, pr, redemption, frequency or [basis] arguments. That is, we provided  $\text{rate} < 0$ ;  $\text{pr} \leq 0$ ;  $\text{redemption} \leq 0$ ; frequency is any number other than 1, 2 or 4; or [basis] is any number other than 0, 1, 2, 3 or 4.
2. #VALUE! error – Occurs when:
  - Any of the arguments provided is non-numeric.
  - The settlement and maturity dates provided are not valid dates.
3. The result from the Excel RATE function appears to be the value 0 or appears as a percentage but shows no decimal places. The problem is often due to the formatting of the cell containing the function. If this is the case, fix the problem by formatting the cell to show a percentage with decimal places.
4. The settlement date is the date a buyer purchases a coupon, such as a bond. The maturity date is the date when a coupon expires. For example, a 30-year bond is issued on January 1, 2010 and is purchased by a buyer six months later. The issue date would be January 1, 2010, the settlement date would be July 1, 2010, and the maturity date would be January 1, 2040, which is 30 years after the January 1, 2010 issue date.

[Click here to download the sample Excel file](#)

## Information

**ERROR.TYPE** Test for a specific error value

error\_val

### What is the ERROR.TYPE Function?

The ERROR.TYPE Function is categorized as an **Information function**. The function returns a number that corresponds to a specific error value. If there is no error, it will return #N/A.

While doing **financial analysis**, we can use this to test a specific error value. Also, we can use ERROR.TYPE to display a customized error message. This can be done using the IF function. The IF function can be used to test for an error value and return a text string, such as a message, instead of the error value.

### Formula

**=ERROR.TYPE(error\_val)**

Where:

**Error\_value** (required argument) – It is the error value for whom we wish to find the identifying number. Though error\_val can be the actual error value, it will usually be a reference to a cell containing a formula that we want to test.

The numbers provided by Excel:

| If error_val is | ERROR.TYPE returns |
|-----------------|--------------------|
| #NULL!          | 1                  |
| #DIV/0!         | 2                  |
| #VALUE!         | 3                  |
| #REF!           | 4                  |
| #NAME?          | 5                  |
| #NUM!           | 6                  |
| #N/A            | 7                  |
| #N/A            | 8                  |
| Anything else   | #N/A               |



## How to use the ERROR.TYPE Function in Excel?

Let's see a few examples to understand how the ERROR.TYPE function works.

### Example

Suppose we wish to display customized messages for certain errors:

|   | A | B                          | C |
|---|---|----------------------------|---|
| 1 |   |                            |   |
| 2 |   | <b>ERROR.TYPE Function</b> |   |
| 3 |   |                            |   |
| 4 |   | #DIV/0!                    |   |
| 5 |   | #NULL!                     |   |
| 6 |   | DATA                       |   |
| 7 |   |                            |   |

Now we wish to input a customized message. What we want is that this formula should check the given cell reference, i.e. cell A2, to see whether the cell contains either the #NULL! error value or the #DIV/0! error value.

If errors are found, the number for the error value is used in the CHOOSE worksheet function to display one of two messages; otherwise, the #N/A error value is returned.

The formula to use is:

**=IF(ERROR.TYPE(A2)<3,CHOOSE(ERROR.TYPE(A2),"Ranges do not intersect","The divisor is zero"))**

| SUM | A | B                          | C  |
|-----|---|----------------------------|--|
|     |   |                            |  |
|     |   | <b>ERROR.TYPE Function</b> |  |
|     |   |                            |  |
|     |   | #DIV/0!                    | The divisor is zero  |
|     |   | #NULL!                     | =IF(ERROR.TYPE(B5)<3,CHOOSE(ERROR.TYPE(B5),"Ranges do not intersect","The divisor is zero")) |
|     |   | DATA                       | #N/A   |
|     |   |                            |  |

We get the result below:

| C4 | A | B                          | C                       |
|----|---|----------------------------|-------------------------|
|    |   |                            |                         |
|    |   | <b>ERROR.TYPE Function</b> |                         |
|    |   |                            |                         |
|    |   | #DIV/0!                    | The divisor is zero     |
|    |   | #NULL!                     | Ranges do not intersect |
|    |   | DATA                       | #N/A                    |
|    |   |                            |                         |

[Click here to download the sample Excel file](#)

**ISBLANK**

Test if a cell is blank

value

**What is the ISBLANK Function?**

The Excel ISBLANK function is categorized under **Information functions**. It helps extract information from a cell. ISBLANK checks a specified cell and tells us if it is blank or not. If it is blank, it will return TRUE; else, it will return FALSE. The function was introduced in MS Excel 2007.

In **financial analysis**, we deal with data all the time. The ISBLANK function is useful in checking if a cell is blank or not. For example, if A5 contains a formula that returns an empty string "" as result, the function will return FALSE. Thus, it helps in removing both regular and non-breaking space characters.

However, if a cell contains good data, as well as non-breaking spaces, it is possible to remove the non-breaking spaces from the data.

**Formula****=ISBLANK(value)**

Where:

- **Value** (required argument) is the value that we wish to test.

**How to use the Excel ISBLANK Function**

As a worksheet function, ISBLANK can be entered as part of a formula in a cell of a worksheet. To understand the uses of this function, let us consider a few examples:

### Example 1

Suppose we are given the following data:

The screenshot shows an Excel spreadsheet with a data table and a conditional formatting rule. The data table is as follows:

|  |     |     |     |     |     |     |
|--|-----|-----|-----|-----|-----|-----|
|  |     |     |     |     |     |     |
|  |     |     |     |     |     |     |
|  |     |     |     |     |     |     |
|  | 101 | 102 | 103 | 104 | 105 | 106 |
|  | 102 | 102 | 104 |     | 106 | 107 |
|  | 103 |     | 105 | 106 | 107 |     |
|  |     | 102 | 106 | 107 |     | 109 |
|  | 105 |     |     | 108 | 109 |     |
|  | 106 | 102 | 108 | 109 | 110 | 111 |
|  |     | 102 | 109 | 110 | 111 | 112 |
|  | 108 | 102 | 110 | 110 |     | 113 |

Below the data table, a conditional formatting rule is applied to the range A2:F9. The rule is named "ISBLANK Function" and is based on the formula `=ISBLANK(A2:F9)`.

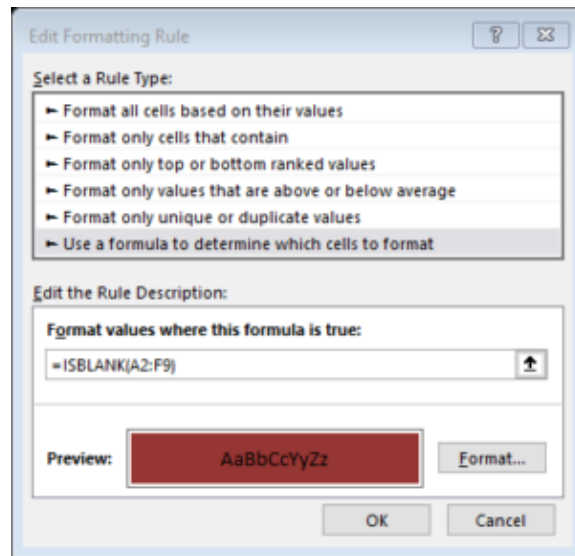
Suppose we wish to highlight cells that are empty with conditional formatting, we can do so using the ISBLANK function. For example, if we want to highlight blank cells in the range A2:F9, we just need to select the range and create a conditional formatting rule based on this formula: `=ISBLANK(A2:F9)`.

How to do conditional formatting?

Under Home tab – Styles, click on Conditional Formatting. Then click on New Rule and then select – Use a Formula to determine which cells to format:

The screenshot shows the "New Formatting Rule" dialog box in Excel. The "Select a Rule Type:" section is expanded to show the option "Use a formula to determine which cells to format". The "Edit the Rule Description:" section shows the text "Format values where this formula is true:" and a text box for entering the formula. The "Preview:" section shows "No Format Set" and a "Format..." button. The "OK" and "Cancel" buttons are at the bottom.

The input formula is shown below:



We will get the results below.

Conditional formatting didn't highlight cell E5. After checking, there is a formula inserted into the cell.

|    | A | B                       | C   | D   | E   | F   | G   | H |
|----|---|-------------------------|-----|-----|-----|-----|-----|---|
| 1  |   |                         |     |     |     |     |     |   |
| 2  |   | <b>ISBLANK Function</b> |     |     |     |     |     |   |
| 3  |   |                         |     |     |     |     |     |   |
| 4  |   | 101                     | 102 | 103 | 104 | 105 | 106 |   |
| 5  |   | 102                     | 102 | 104 |     | 106 | 107 |   |
| 6  |   | 103                     |     | 105 | 106 | 107 |     |   |
| 7  |   |                         | 102 | 106 | 107 |     | 109 |   |
| 8  |   | 105                     |     |     | 108 | 109 |     |   |
| 9  |   | 106                     | 102 | 108 | 109 | 110 | 111 |   |
| 10 |   |                         | 102 | 109 | 110 | 111 | 112 |   |
| 11 |   | 108                     | 102 | 110 | 110 |     | 113 |   |
| 12 |   |                         |     |     |     |     |     |   |

The Excel ISBLANK function will return TRUE when a cell is actually empty. If a cell contains a formula that returns an empty string (i.e. ""), ISBLANK won't see these cells as blank, and won't return TRUE, so they won't be highlighted as shown above.

When we use a formula to apply conditional formatting, the formula is evaluated relative to the active cell in the selection at the time the rule is created. In this case, the formula =ISBLANK(B4) is evaluated for each cell in B4:G11. As B4 is entered as a relative address, the address will be updated each time the formula is applied, and ISBLANK() is run on each cell in the range.

## Example 2

Suppose we wish to get the first non-blank value (text or number) in a one-row range, we can use an array formula based on the INDEX, MATCH, and ISBLANK functions.

We are given the data below:

|   | A | B                | C       | D       | E | F       | G | H |
|---|---|------------------|---------|---------|---|---------|---|---|
| 1 |   |                  |         |         |   |         |   |   |
| 2 |   | ISBLANK Function |         |         |   |         |   |   |
| 3 |   |                  |         |         |   |         |   |   |
| 4 |   |                  | Peaches | Oranges |   | Mangoes |   |   |
| 5 |   |                  |         |         |   |         |   |   |

Here, we want to get the first non-blank cell, but we don't have a direct way to do that in Excel. We could use VLOOKUP with a wildcard \*, but that will only work for text, not numbers.

Hence, we need to build the functionality by nesting formulas. One way to do it is to use an array function that "tests" cells and returns an array of TRUE/FALSE values that we can then match with MATCH. Now MATCH looks for FALSE inside the array and returns the position of the first match found, which, in this case, is 2. Now the INDEX function takes over and gets the value at position 2 in the array, which, in this case, is the value PEACHES.

As this is an array formula, we need to enter it with CTRL+SHIFT+ENTER.

|   | A | B                | C       | D       | E | F       | G   | H | I | J | K | L |
|---|---|------------------|---------|---------|---|---------|---|---|---|---|---|---|
| 1 |   |                  |         |         |   |         |   |   |   |   |   |   |
| 2 |   | ISBLANK Function |         |         |   |         |   |   |   |   |   |   |
| 3 |   |                  |         |         |   |         |   |   |   |   |   |   |
| 4 |   |                  | Peaches | Oranges |   | Mangoes |   |   |   |   |   |   |
| 5 |   |                  |         |         |   |         |   |   |   |   |   |   |
| 6 |   |                  |         |         |   |         |   |   |   |   |   |   |
| 7 |   |                  |         |         |   |         | {=INDEX(B4:G4,MATCH(FALSE,ISBLANK(B4:G4),0))} |   |   |   |   |   |
| 8 |   |                  |         |         |   |         |   |   |   |   |   |   |

We get the results below:

|   | A | B                | C       | D       | E | F       | G       | H | I |  |
|---|---|------------------|---------|---------|---|---------|---------|---|---|--|
| 1 |   |                  |         |         |   |         |         |   |   |  |
| 2 |   | ISBLANK Function |         |         |   |         |         |   |   |  |
| 3 |   |                  |         |         |   |         |         |   |   |  |
| 4 |   |                  | Peaches | Oranges |   | Mangoes |         |   |   |  |
| 5 |   |                  |         |         |   |         |         |   |   |  |
| 6 |   |                  |         |         |   |         |         |   |   |  |
| 7 |   |                  |         |         |   |         | Peaches |   |   |  |
| 8 |   |                  |         |         |   |         |         |   |   |  |
| 9 |   |                  |         |         |   |         |         |   |   |  |

[Click here to download the sample Excel file.](#)

## Logical

**AND** Test multiple conditions with AND

logical1

logical2

...

### What is the AND Function?

The AND function is categorized under **Logical functions**. It is used to determine if the given conditions in a test are TRUE. For example, we can use the function to test if a number in cell B1 is greater than 100 and less than 50.

As a **financial analyst**, the function is useful in testing multiple conditions, specifically, if multiple conditions are met to make sure they all are true. Also, it helps us avoid extra nested IFS, and the AND can be combined with the OR function.

### Formula

**=AND(logical1, [logical2], ...)**

The AND function uses the following arguments:

1. **Logical1** (required argument) – It is the first condition or logical value to be evaluated.
2. **Logical2** (optional requirement) – It is the second condition or logical value to be evaluated.

### Notes:

1. We can set up to 255 conditions for the AND function.
2. The function tests a number of supplied conditions and returns:
  1. TRUE if ALL of the conditions evaluate to TRUE; or
  2. FALSE otherwise (i.e. if ANY of the conditions evaluate to FALSE).

### How to use the AND Function in Excel?

To understand the uses of the AND function, let us consider a few examples:

### Example 1

Suppose we wish to calculate the bonus for every salesperson in our company. To be eligible for a 5% bonus, the salesperson should have achieved sales higher than \$5,000 in a year. Otherwise, they should have achieved an account goal of 4 accounts or higher. To earn a bonus of 15%, they should have achieved both conditions together.

We are given the data given below:

|   | A | B                   | C                  | D               |
|---|---|---------------------|--------------------|-----------------|
| 1 |   |                     |                    |                 |
| 2 |   | <b>AND Function</b> |                    |                 |
| 3 |   | <b>Salesperson</b>  | <b>Total sales</b> | <b>Accounts</b> |
| 4 |   | Jim                 | 4500               | 3               |
| 5 |   | William             | 5600               | 5               |
| 6 |   | Lily                | 6000               | 8               |
| 7 |   | Robot               | 6500               | 4               |

To calculate the bonus commission, we will use the AND function. The formula will be:

|   |   | IRR                              |                    |                 |              |                                  |
|---|---|----------------------------------|--------------------|-----------------|--------------|----------------------------------|
|   |   | =IF(AND(C4>5000,D4>=4),C4*15%,0) |                    |                 |              |                                  |
|   | A | B                                | C                  | D               | E            | F                                |
| 1 |   |                                  |                    |                 |              |                                  |
| 2 |   | <b>AND Function</b>              |                    |                 |              |                                  |
| 3 |   | <b>Salesperson</b>               | <b>Total sales</b> | <b>Accounts</b> | <b>Bonus</b> | <b>Bonus commission</b>          |
| 4 |   | Jim                              | 4500               | 3               |              | =IF(AND(C4>5000,D4>=4),C4*15%,0) |
| 5 |   | William                          | 5600               | 5               | 280          | 840                              |
| 6 |   | Lily                             | 6000               | 8               | 300          | 900                              |
| 7 |   | Robot                            | 6500               | 4               | 325          | 975                              |
| 8 |   |                                  |                    |                 |              |                                  |

The formula for calculating the bonus does not require the AND function but we need to use the OR function. The formula to use will be:

|   |   | IRR                               |                    |                 |              |                                   |
|---|---|-----------------------------------|--------------------|-----------------|--------------|-----------------------------------|
|   |   | =IF(OR(C4>5000,(D4>4)),(C4*5%),0) |                    |                 |              |                                   |
|   | A | B                                 | C                  | D               | E            | F                                 |
| 1 |   |                                   |                    |                 |              |                                   |
| 2 |   | <b>AND Function</b>               |                    |                 |              |                                   |
| 3 |   | <b>Salesperson</b>                | <b>Total sales</b> | <b>Accounts</b> | <b>Bonus</b> | <b>Bonus commission</b>           |
| 4 |   | Jim                               | 4500               | 3               |              | =IF(OR(C4>5000,(D4>4)),(C4*5%),0) |
| 5 |   | William                           | 5600               | 5               | 280          | 840                               |
| 6 |   | Lily                              | 6000               | 8               | 300          | 900                               |
| 7 |   | Robot                             | 6500               | 4               | 325          | 975                               |

The resulting table will be similar to the one below:

The screenshot shows an Excel spreadsheet with the following data:

|   | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 |   |   |   |   |   |   |   |
| 2 |   |   |   |   |   |   |   |
| 3 |   |   |   |   |   |   |   |
| 4 |   |   |   |   |   |   |   |
| 5 |   |   |   |   |   |   |   |
| 6 |   |   |   |   |   |   |   |
| 7 |   |   |   |   |   |   |   |
| 8 |   |   |   |   |   |   |   |

The formula bar shows the formula: `=IF(OR(C4>5000,(D4>4)),(C4*5%),0)`

The table data is as follows:

| Salesperson | Total sales | Accounts | Bonus | Bonus commission |
|-------------|-------------|----------|-------|------------------|
| Jim         | 4500        | 3        | 0     | 0                |
| William     | 5600        | 5        | 280   | 840              |
| Lily        | 6000        | 8        | 300   | 900              |
| Robot       | 6500        | 4        | 325   | 975              |



## Example 2

Let's see how we can use the AND function to test if a numeric value falls between two specific numbers. For this, we can use the AND function with two logical tests.

Suppose we are given the data below:

|   | A | B                   | C               | D               |
|---|---|---------------------|-----------------|-----------------|
| 1 |   |                     |                 |                 |
| 2 |   | <b>AND Function</b> |                 |                 |
| 3 |   | <b>Number 1</b>     | <b>Number 2</b> | <b>Number 3</b> |
| 4 |   | 25                  | 15              | 65              |
| 5 |   | 7                   | 25              | 55              |
| 6 |   | 100                 | 102             | 106             |
| 7 |   | 5                   | 15              | 10              |

The formula to use is:

|   | A | B                   | C               | D               | E                                 | F | G | H |
|---|---|---------------------|-----------------|-----------------|-----------------------------------|---|---|---|
| 1 |   |                     |                 |                 |                                   |   |   |   |
| 2 |   | <b>AND Function</b> |                 |                 |                                   |   |   |   |
| 3 |   | <b>Number 1</b>     | <b>Number 2</b> | <b>Number 3</b> |                                   |   |   |   |
| 4 |   | 25                  | 15              | 65              | =AND(D4>MIN(B4,C4),D4<MAX(B4,C4)) |   |   |   |
| 5 |   | 7                   | 25              | 55              | FALSE                             |   |   |   |
| 6 |   | 100                 | 102             | 106             | FALSE                             |   |   |   |
| 7 |   | 5                   | 15              | 10              | TRUE                              |   |   |   |
| 8 |   |                     |                 |                 |                                   |   |   |   |
| 9 |   |                     |                 |                 |                                   |   |   |   |

In the above formula, the value is compared to the smaller of the two numbers, determined by the MIN function. In the second expression, the value is compared to the larger of the two numbers, determined by the MAX function. The AND function will return TRUE only when the value is greater than the smaller number and less than the larger number.

We get the results below:

|   | A | B                   | C               | D               | E     | F | G | H |
|---|---|---------------------|-----------------|-----------------|-------|---|---|---|
| 1 |   |                     |                 |                 |       |   |   |   |
| 2 |   | <b>AND Function</b> |                 |                 |       |   |   |   |
| 3 |   | <b>Number 1</b>     | <b>Number 2</b> | <b>Number 3</b> |       |   |   |   |
| 4 |   | 25                  | 15              | 65              | FALSE |   |   |   |
| 5 |   | 7                   | 25              | 55              | FALSE |   |   |   |
| 6 |   | 100                 | 102             | 106             | FALSE |   |   |   |
| 7 |   | 5                   | 15              | 10              | TRUE  |   |   |   |
| 8 |   |                     |                 |                 |       |   |   |   |
| 9 |   |                     |                 |                 |       |   |   |   |

**Things to remember about the AND Function**

1. VALUE! error – Occurs when no logical values are found or created during the evaluation.
2. Test values or empty cells provided as arguments are ignored by the AND function.

[Click here to download the sample Excel file](#)

**IF** Test for a specific condition    `logical_test`    `value_if_true`    `value_if_false`

### What is an Excel IF Statement?

An Excel IF Statement tests a given condition and returns one value for a TRUE result, and another value for a FALSE result. For example, if sales total more than \$5,000 then return a "Yes" for Bonus, otherwise, return a "No" for Bonus. We can also use the IF function to evaluate a single function or we can include several IF functions in one formula. Multiple IF statements in Excel are known as nested IF statements.

## Excel IF Statement

---

= IF (Cell C2 >= Cell D2, "Yes it is", "No it isn't")

i.e.

When C2 is 9 and D2 is 7, the result = "Yes it is"

When C2 is 3 and D2 is 5, the result = "No it isn't"

As a **financial analyst**, the IF function is used often to evaluate and analyze data by evaluating specific conditions.

The function can be used to evaluate text, values, and even errors. It is not limited to only checking if one thing is equal to another and returning a single result. We can also use mathematical operators and perform additional calculations depending on our criteria. We can also nest multiple IF functions together to perform multiple comparisons.

## IF Formula

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

The formula uses the following arguments:

1. **Logical\_test** (required argument) – It is the condition to be tested and evaluated as either TRUE or FALSE.
2. **Value\_if\_true** (optional argument) – It is the value that will be returned if the logical\_test evaluates to TRUE.
3. **Value\_if\_false** (optional argument) – It is the value that will be returned if the logical\_test evaluates to FALSE.

When using the IF function to construct a test, we can use following logical operators:

- = (equal to)
- > (greater than)
- >= (greater than or equal to)
- < (less than)
- <= (less than or equal to)
- <> (not equal to)

## How to use the Excel IF Function?

To understand the uses of the Excel IF statement function, let us consider a few examples:

### Example 1 – Simple Excel IF Statement

Suppose we wish to do a very simple test. We want to test if the value in cell C2 of greater than or equal to the value in cell D2. If the argument is true, then we want to return some text stating “Yes it is”, and if it’s not true then we want to display “No it isn’t”.

You can see exactly how the Excel IF statement works in simple example below.

#### Result when true:

|    | A          | B                                     | C         | D                | E |
|----|------------|---------------------------------------|-----------|------------------|---|
| 1  |            |                                       |           |                  |   |
| 2  |            |                                       | 9         | 7                |   |
| 3  |            |                                       |           |                  |   |
| 4  |            |                                       |           |                  |   |
| 5  | formula -> | =IF(C2>=D2,"Yes it is","No it isn't") |           |                  |   |
| 6  |            |                                       |           |                  |   |
| 7  |            |                                       | result -> | <b>Yes it is</b> |   |
| 8  |            |                                       |           |                  |   |
| 9  |            |                                       |           |                  |   |
| 10 |            |                                       |           |                  |   |

#### Result when false:

|    | A          | B                                     | C         | D                  | E |
|----|------------|---------------------------------------|-----------|--------------------|---|
| 1  |            |                                       |           |                    |   |
| 2  |            |                                       | 3         | 5                  |   |
| 3  |            |                                       |           |                    |   |
| 4  |            |                                       |           |                    |   |
| 5  | formula -> | =IF(C2>=D2,"Yes it is","No it isn't") |           |                    |   |
| 6  |            |                                       |           |                    |   |
| 7  |            |                                       | result -> | <b>No it isn't</b> |   |
| 8  |            |                                       |           |                    |   |
| 9  |            |                                       |           |                    |   |
| 10 |            |                                       |           |                    |   |

[Download the simple XLS template.](#)

## Example 2 – Excel IF Statement

Suppose we wish to test a cell and ensure that an action is taken if the cell is not blank. We are given the data below:

|    | A | B  | C             | D              | E |
|----|---|--|---------------|----------------|---|
| 1  |   |  |               |                |   |
| 2  |   | <b>IF Function</b>                                   |               |                |   |
| 3  |   |  |               |                |   |
| 4  |   | <b>AGM Preparation list</b>                          | <b>Status</b> | <b>Remarks</b> |   |
| 5  |   | Directors report to be finalised and sent for review | Closed        | 01/01/18       |   |
| 6  |   | Finalize Annual report                               | Open          |                |   |
| 7  |   | AGM Notice   | Closed        | 01/15/18       |   |
| 8  |   | Prepare attendance register                          | Open          |                |   |
| 9  |   | Ready the documents needed                           | Open          |                |   |
| 10 |   | Follow up with Auditors                              | Open          |                |   |
| 11 |   |  |               |                |   |

In the worksheet above, we listed AGM-related tasks in Column A. Remarks contain the date of completion. In Column B, we will use a formula to check if cells in Column C are empty or not. If a cell is blank, the formula will assign a status "open." However, if a cell contains a date then the formula will assign a status as "closed." The formula used is:

|    | A | B  | C                             | D              | E |
|----|---|--|-------------------------------|----------------|---|
| 1  |   |  |                               |                |   |
| 2  |   | <b>IF Function</b>                                   |                               |                |   |
| 3  |   |  |                               |                |   |
| 4  |   | <b>AGM Preparation list</b>                          | <b>Status</b>                 | <b>Remarks</b> |   |
| 5  |   | Directors report to be finalised and sent for review | =IF(D5<>"", "Closed", "Open") |                |   |
| 6  |   | Finalize Annual report                               | Open                          |                |   |
| 7  |   | AGM Notice   | Closed                        | 01/15/18       |   |
| 8  |   | Prepare attendance register                          | Open                          |                |   |
| 9  |   | Ready the documents needed                           | Open                          |                |   |
| 10 |   | Follow up with Auditors                              | Open                          |                |   |

We get the results below:

The screenshot shows an Excel spreadsheet with the following data:

|    | A | B  | C             | D              |
|----|---|--|---------------|----------------|
| 1  |   |  |               |                |
| 2  |   | <b>IF Function</b>                                   |               |                |
| 3  |   |  |               |                |
| 4  |   | <b>AGM Preparation list</b>                          | <b>Status</b> | <b>Remarks</b> |
| 5  |   | Directors report to be finalised and sent for review | Closed        | 01/01/18       |
| 6  |   | Finalize Annual report                               | Open          |                |
| 7  |   | AGM Notice   | Closed        | 01/15/18       |
| 8  |   | Prepare attendance register                          | Open          |                |
| 9  |   | Ready the documents needed                           | Open          |                |
| 10 |   | Follow up with Auditors                              | Open          |                |
| 11 |   |  |               |                |

The formula bar at the top shows the formula: `=IF(D5<>,"","Closed","Open")`

### Example 3 – Excel IF Statement

Generally, sellers provide a discount based on quantity purchased. Suppose we are given the following data:

|   | A | B                    | C                     | D |
|---|---|----------------------|-----------------------|---|
| 1 |   |                      |                       |   |
| 2 |   | IF Function          |                       |   |
| 3 |   |                      |                       |   |
| 4 |   | <b>Unit quantity</b> | <b>Price per unit</b> |   |
| 5 |   | 1 to 10              | \$30                  |   |
| 6 |   | 11 to 25             | \$28                  |   |
| 7 |   | 26 to 65             | \$25                  |   |
| 8 |   | 65 and above         | \$22                  |   |

Using multiple IF functions, we can create a formula to check multiple conditions and perform different calculations depending on what amount range the specified quantity falls in. To calculate the total price for 100 items, the formula will be:

IRR    :    X    ✓    fx    =C11\*IF(C11>=65,22,IF(C11>=26,25,IF(C11>=11,28,IF(C11>=1,30,""))))

|    | A | B                    | C  | D | E | F | G | H | I | J |
|----|---|----------------------|--|---|---|---|---|---|---|---|
| 1  |   |                      |  |   |   |   |   |   |   |   |
| 2  |   | IF Function          |  |   |   |   |   |   |   |   |
| 3  |   |                      |  |   |   |   |   |   |   |   |
| 4  |   | <b>Unit quantity</b> | <b>Price per unit</b>  |   |   |   |   |   |   |   |
| 5  |   | 1 to 10              | \$30   |   |   |   |   |   |   |   |
| 6  |   | 11 to 25             | \$28   |   |   |   |   |   |   |   |
| 7  |   | 26 to 65             | \$25   |   |   |   |   |   |   |   |
| 8  |   | 65 and above         | \$22   |   |   |   |   |   |   |   |
| 9  |   |                      |  |   |   |   |   |   |   |   |
| 10 |   |                      |  |   |   |   |   |   |   |   |
| 11 |   | Quantity             | 75   |   |   |   |   |   |   |   |
| 12 |   | Total                | =C11*IF(C11>=65,22,IF(C11>=26,25,IF(C11>=11,28,IF(C11>=1,30,"")))) |   |   |   |   |   |   |   |

We get the result below:

C12    :    X    ✓    fx    =C11\*IF(C11>=65,22,IF(C11>=26,25,IF(C11>=11,28,IF(C11>=1,30,""))))

|    | A | B                    | C                     | D | E | F | G | H | I | J |
|----|---|----------------------|-----------------------|---|---|---|---|---|---|---|
| 1  |   |                      |                       |   |   |   |   |   |   |   |
| 2  |   | IF Function          |                       |   |   |   |   |   |   |   |
| 3  |   |                      |                       |   |   |   |   |   |   |   |
| 4  |   | <b>Unit quantity</b> | <b>Price per unit</b> |   |   |   |   |   |   |   |
| 5  |   | 1 to 10              | \$30                  |   |   |   |   |   |   |   |
| 6  |   | 11 to 25             | \$28                  |   |   |   |   |   |   |   |
| 7  |   | 26 to 65             | \$25                  |   |   |   |   |   |   |   |
| 8  |   | 65 and above         | \$22                  |   |   |   |   |   |   |   |
| 9  |   |                      |                       |   |   |   |   |   |   |   |
| 10 |   |                      |                       |   |   |   |   |   |   |   |
| 11 |   | Quantity             | 75                    |   |   |   |   |   |   |   |
| 12 |   | Total                | 1,650                 |   |   |   |   |   |   |   |



### Things to remember about the IF Function

1. The Excel IF function will work if the logical\_test returns a numeric value. In such case, any non-zero value is treated as TRUE and zero is treated as FALSE.
2. #VALUE! error – Occurs when the given logical\_test argument cannot be evaluated as TRUE or FALSE.
3. When any of the arguments is provided to the function as arrays, the IF function will evaluate every element of the array.
4. If we wish to count conditions, we should use the COUNTIF and COUNTIFS functions.
5. If we wish to add up conditions, we should use the SUMIF and SUMIFS functions.

### Reasons to use an Excel IF Statement

There are many reasons why an analyst or anyone who uses Excel would want to build IF formulas.

Common examples include:

- To test if an argument is true or false
- To output a NUMBER
- To output some TEXT
- To generate a conditional formula (i.e. the result is C3+B4 if true and N9-E5 if false)
- To create scenarios to be used in financial modeling
- To calculate a debt schedule or fixed asset depreciation schedule in accounting

[Click here to download the sample Excel file](#)

**IFERROR**

Trap and handle errors

value

value\_if\_error

**What is the IFERROR Function?**

The IFERROR Function is categorized under **Logical Functions**. IFERROR belongs to a group of error-checking functions such as ISERR, ISERROR, IFERROR, and ISNA. The function will return a customized result when the result of the formula is an error and a standard result if there is no error.

In **financial analysis**, we need to deal with formulas and data. Often, we will get errors in cells with a formula. IFERROR is used to trap and handle errors produced by other formulas and functions. It handles errors such as #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!.

For example, the formula in C2 is A2/B2, where:

|   | A  | B | C       |
|---|----|---|---------|
| 1 |    |   |         |
| 2 | 25 | 0 | #DIV/0! |

Instead of the resulting error, we can use IFERROR to return a customized message such as "Invalid input."

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

The IFERROR Function uses the following arguments:

1. **Value** (required argument) – It is the expression or value that needs to be tested. It is generally provided as a cell address.
2. **Value\_if\_error** (required argument) – The value that will be returned if the formula evaluates to an error.

To learn more, launch our [free Excel crash course](#) now!

**How to use the IFERROR Function in Excel?**

As a worksheet function, IFERROR can be entered as part of a formula in a cell of a worksheet.

To understand the uses of the function, let's consider a few examples:

### Example 1

Suppose we are given the following data:

|    | A | B                       | C                          | D                  | E                   |
|----|---|-------------------------|----------------------------|--------------------|---------------------|
| 1  |   |                         |                            |                    |                     |
| 2  |   | <b>IFERROR Function</b> |                            |                    |                     |
| 3  |   |                         |                            |                    |                     |
| 4  |   | <b>Name of Student</b>  | <b>Marks in Statistics</b> | <b>Total Marks</b> | <b>%Total Marks</b> |
| 5  |   | Amy                     | 42                         | #REF!              | #REF!               |
| 6  |   | Wilson                  | 35                         | #DIV/0!            | #DIV/0!             |
| 7  |   | William                 | 12                         | 150.00             | 0.08                |
| 8  |   | Sabrina                 | 36                         | #NAME?             | #NAME?              |
| 9  |   | Tom                     | 65                         | #NULL!             | #NULL!              |
| 10 |   | Jane                    | 25                         | 150.00             | 0.16667             |

Using the ISERROR formula, we can remove these errors. We will put a customized message – “Invalid data.” The formula to be used is:

|   | A | B                       | C                          | D                  | E                   | F                               | G | H |
|---|---|-------------------------|----------------------------|--------------------|---------------------|---------------------------------|---|---|
| 1 |   |                         |                            |                    |                     |                                 |   |   |
| 2 |   | <b>IFERROR Function</b> |                            |                    |                     |                                 |   |   |
| 3 |   |                         |                            |                    |                     |                                 |   |   |
| 4 |   | <b>Name of Student</b>  | <b>Marks in Statistics</b> | <b>Total Marks</b> | <b>%Total Marks</b> | <b>%Total Marks</b>             |   |   |
| 5 |   | Amy                     | 42                         | #REF!              | #REF!               | =IFERROR(C5/D5,"Invalid entry") |   |   |

We will get the result below:

|    | A | B                       | C                          | D                  | E                   | F                   | G |
|----|---|-------------------------|----------------------------|--------------------|---------------------|---------------------|---|
| 1  |   |                         |                            |                    |                     |                     |   |
| 2  |   | <b>IFERROR Function</b> |                            |                    |                     |                     |   |
| 3  |   |                         |                            |                    |                     |                     |   |
| 4  |   | <b>Name of Student</b>  | <b>Marks in Statistics</b> | <b>Total Marks</b> | <b>%Total Marks</b> | <b>%Total Marks</b> |   |
| 5  |   | Amy                     | 42                         | #REF!              | #REF!               | Invalid entry       |   |
| 6  |   | Wilson                  | 35                         | #DIV/0!            | #DIV/0!             | Invalid entry       |   |
| 7  |   | William                 | 12                         | 150.00             | 0.08                | 8%                  |   |
| 8  |   | Sabrina                 | 36                         | #NAME?             | #NAME?              | Invalid entry       |   |
| 9  |   | Tom                     | 65                         | #NULL!             | #NULL!              | Invalid entry       |   |
| 10 |   | Jane                    | 25                         | 150.00             | 0.16667             | 17%                 |   |

## Example 2

The most common use of IFERROR function is with VLOOKUP function, where it is used to inform users that the value they are looking out for is not there.

Suppose we operate a restaurant and maintain an inventory of a few vegetables and fruits. Now we compare it with the list of items we want according to the day's menu. The results look like this:

|    | A | B                | C            | D                            | E | F                   | G    |  |
|----|---|------------------|--------------|------------------------------|---|---------------------|------|--|
| 1  |   |                  |              |                              |   |                     |      |  |
| 2  |   | IFERROR Function |              |                              |   |                     |      |  |
| 3  |   |                  |              |                              |   |                     |      |  |
| 4  |   | <b>Item</b>      | <b>Price</b> | <b>Quantity bought (kgs)</b> |   | <b>W/o If Error</b> |      |  |
| 5  |   | Oranges          | 12           | 1                            |   | Oranges             | 1    |  |
| 6  |   | Peaches          | 23           | 5                            |   | Apples              | 2    |  |
| 7  |   | Apples           | 32           | 2                            |   | Pineapple           | #N/A |  |
| 8  |   | Spinach          | 12           | 4                            |   |                     |      |  |
| 9  |   | Avacado          | 25           | 5                            |   |                     |      |  |
| 10 |   |                  |              |                              |   |                     |      |  |
| 11 |   |                  |              |                              |   |                     |      |  |

Instead of the #N/A error message, we can input a customized message. It will look better when we are using a long list.

The formula to use will be:

|   | A | B                | C            | D                            | E | F         | G    | H   | I | J | K | L | M |
|---|---|------------------|--------------|------------------------------|---|-----------|------|---|---|---|---|---|---|
| 1 |   |                  |              |                              |   |           |      |   |   |   |   |   |   |
| 2 |   | IFERROR Function |              |                              |   |           |      |   |   |   |   |   |   |
| 3 |   |                  |              |                              |   |           |      |   |   |   |   |   |   |
| 4 |   | <b>Item</b>      | <b>Price</b> | <b>Quantity bought (kgs)</b> |   |           |      | <b>W/o If Error with if Error</b>                       |   |   |   |   |   |
| 5 |   | Oranges          | 12           | 1                            |   | Oranges   | 1    | =IFERROR(VLOOKUP(F5,\$B\$5:\$D\$9,3,FALSE),"Not found") |   |   |   |   |   |
| 6 |   | Peaches          | 23           | 5                            |   | Apples    | 2    |   |   |   |   |   |   |
| 7 |   | Apples           | 32           | 2                            |   | Pineapple | #N/A | Not found   |   |   |   |   |   |
| 8 |   | Spinach          | 12           | 4                            |   |           |      |   |   |   |   |   |   |
| 9 |   | Avacado          | 25           | 5                            |   |           |      |   |   |   |   |   |   |

We will get the result below:

|   | A | B                | C            | D                            | E | F                                 | G    | H         |
|---|---|------------------|--------------|------------------------------|---|-----------------------------------|------|-----------|
| 1 |   |                  |              |                              |   |                                   |      |           |
| 2 |   | IFERROR Function |              |                              |   |                                   |      |           |
| 3 |   |                  |              |                              |   |                                   |      |           |
| 4 |   | <b>Item</b>      | <b>Price</b> | <b>Quantity bought (kgs)</b> |   | <b>W/o If Error with if Error</b> |      |           |
| 5 |   | Oranges          | 12           | 1                            |   | Oranges                           | 1    | 1         |
| 6 |   | Peaches          | 23           | 5                            |   | Apples                            | 2    | 2         |
| 7 |   | Apples           | 32           | 2                            |   | Pineapple                         | #N/A | Not found |
| 8 |   | Spinach          | 12           | 4                            |   |                                   |      |           |
| 9 |   | Avacado          | 25           | 5                            |   |                                   |      |           |

Suppose we are given monthly sales data for the first quarter. We create three worksheets – Jan, Feb, March.

Now we wish to find out the sales figures of certain order ids. As data would be derived by VLOOKUP from three sheets, using IFERROR, we can use the following formula:

```
=IFERROR(VLOOKUP(A2,'JAN'!A2:B5,2,0),IFERROR(VLOOKUP(A2,'FEB'!A2:B5,2,0),IFERROR(VLOOKUP(A2,'MAR'!A2:B5,2,0),"not found")))
```

### Example 3

Let's see how to use IFERROR with an array formula. When we supply an array formula or expression that results in an array in the value argument of the IFERROR function, it will return an array of values for each cell in the specified range.

Suppose we are given the following data:

|    | A | B                       | C                  | D                     | E     |
|----|---|-------------------------|--------------------|-----------------------|-------|
| 1  |   |                         |                    |                       |       |
| 2  |   | <b>IFERROR Function</b> |                    |                       |       |
| 3  |   |                         |                    |                       |       |
| 4  |   | <b>Item</b>             | <b>Total Amoun</b> | <b>Price per unit</b> |       |
| 5  |   | Apple                   | 50,000             | 10                    | 5,000 |
| 6  |   | Mango                   | 50,000             | 100                   | 500   |
| 7  |   | Pineapple               | 50,000             | 20                    | 2,500 |
| 8  |   | Avacado                 | 50,000             | 0                     |       |
| 9  |   |                         |                    |                       |       |
| 10 |   |                         |                    |                       |       |

To calculate total quantity, we need to divide Total Amount with Price per unit. We can use an array formula, which divides each cell in the range B2:B4 by the corresponding cell of the range C2:C4, and then adds up the results:

`=SUM($B$2:$B$5/$C$2:$C$5)`

The formula will work perfectly unless the divisor range does not contain zeros or empty cells. If there is at least one 0 value or blank cell, the #DIV/0! error is returned:

|   | A | B                       | C                  | D                     | E     | F | G                     | H |
|---|---|-------------------------|--------------------|-----------------------|-------|---|-----------------------|---|
| 1 |   |                         |                    |                       |       |   |                       |   |
| 2 |   | <b>IFERROR Function</b> |                    |                       |       |   |                       |   |
| 3 |   |                         |                    |                       |       |   |                       |   |
| 4 |   | <b>Item</b>             | <b>Total Amoun</b> | <b>Price per unit</b> |       |   | <b>Total quantity</b> |   |
| 5 |   | Apple                   | 50,000             | 10                    | 5,000 |   | #DIV/0!               |   |
| 6 |   | Mango                   | 50,000             | 100                   | 500   |   | 8000                  |   |
| 7 |   | Pineapple               | 50,000             | 20                    | 2,500 |   |                       |   |
| 8 |   | Avacado                 | 50,000             | 0                     |       |   |                       |   |

In order to fix the above error, we can use the IFERROR Function. The new formula would be:

`{=SUM(IFERROR($B$2:$B$5/$C$2:$C$5,0))}`

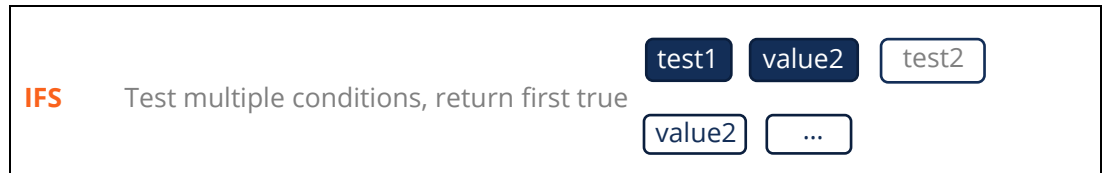
|   | A | B                       | C                   | D                     | E     | F | G                     |
|---|---|-------------------------|---------------------|-----------------------|-------|---|-----------------------|
| 1 |   |                         |                     |                       |       |   |                       |
| 2 |   | <b>IFERROR Function</b> |                     |                       |       |   |                       |
| 3 |   |                         |                     |                       |       |   |                       |
| 4 |   | <b>Item</b>             | <b>Total Amount</b> | <b>Price per unit</b> |       |   | <b>Total quantity</b> |
| 5 |   | Apple                   | 50,000              | 10                    | 5,000 |   | #DIV/0!               |
| 6 |   | Mango                   | 50,000              | 100                   | 500   |   | 8000                  |
| 7 |   | Pineapple               | 50,000              | 20                    | 2,500 |   |                       |
| 8 |   | Avacado                 | 50,000              | 0                     |       |   |                       |

What the formula did here was that it divided the value in column B by a value in column C in each row (50000/10, 50000/100, 50000/20 and 0/0) and return an array of results {5000; 500; 2500; & #DIV/0!}. Thus, IFERROR function identified all #DIV/0! errors and replaced them with zeros. The SUM function then adds up the values in the resulting array {5,000; 500; 2,500; 0} and gives the final result (8000).

### Things to remember about the IFERROR Function

1. The IFERROR Function was introduced in Excel 2007 and is available in all subsequent Excel versions.
2. If the value argument is a blank cell, it is treated as an empty string ("") and not an error.
3. If value is an array formula, IFERROR returns an array of results for each cell in the range specified in value.

[Click here to download the sample Excel file.](#)



## What is the IFS Function?

The IFS Function in Excel is a Logical function that was introduced in Excel 2016. The function is an alternative to the Nested IF function and is much easier to use. The IFS function checks if one or more than one conditions are observed or not and accordingly returns a value that meets the first TRUE condition.

## Formula

= IFS(logical\_test1, Value1 [logical\_test2, Value2] ..., [logical\_test127, Value127])

Where:

1. **Logical\_test1** – First logical test. It is a required argument and is the condition that is used by Excel to evaluate whether it is TRUE or FALSE.
2. **Value1** – Result when logical\_test1 is true. If required, we can keep it empty.

The rest of the logical\_test and Value arguments are optional; the function allows the user to put 127 logical\_test arguments using the IFS function.

## How to use the IFS Function in Excel?

It is a built-in function which can be used as a worksheet function in Excel. Let's take an example:

Let's assume we wish to assign grades to marks earned by students. We can use the IFS Function in the following manner:

|   | A | B | C   | D            | E | F | G | H | I | J |
|---|---|---|---|--------------|---|---|---|---|---|---|
| 1 |   |   |   |              |   |   |   |   |   |   |
| 2 |   |   | <b>IFS Function</b>                                     |              |   |   |   |   |   |   |
| 3 |   |   |   |              |   |   |   |   |   |   |
| 4 |   |   | <u>Marks</u>  | <u>Grade</u> |   |   |   |   |   |   |
| 5 |   |   | =IFS(B5>80,"A",B5>70,"B",B5>60,"C",B5>50,"D",B5>40,"E", |              |   |   |   |   |   |   |
| 6 |   |   | B5>30,"F")  |              |   |   |   |   |   |   |



The formula used is:

`IFS(A2>80,"A",A2>70,"B",A2>60,"C",A2>50,"D",A2>40,"E",A2>30,"F")`, which says that if cell A2 is greater than 80 then return an "A" and so on.

Using this formula, the result would be:

The screenshot shows an Excel spreadsheet with the following data:

| Marks | Grade |
|-------|-------|
| 90    | A     |
| 80    | B     |
| 70    | C     |
| 60    | D     |
| 50    | E     |
| 40    | F     |

The formula bar shows the formula: `=IFS(B5>80,"A",B5>70,"B",B5>60,"C",B5>50,"D",B5>40,"E",B5>30,"F")`

### Examples of the IFS Function in Excel

To understand the uses of this function, let's consider few examples:

### Example 1 - Using IFS with ELSE

Let's assume we have a list of items and we need to classify them under three common headings: Vegetable, Fruit, Green Vegetable and Beverage. When we use the IFS function and give the formula  
`=IFS(A2="Apple","Fruit",B5="Banana","Fruit",A2="Spinach","Green Vegetable",A2="coffee","Beverage",A2="cabbage","Green Vegetable",A2="capsicum","Vegetable")`

The screenshot shows the Excel formula bar with the following formula: `=IFS(B5="Apple","Fruit",B5="Banana","Fruit",B5="Spinach","Green Vegetable",B5="coffee","Beverage",B5="cabbage","Green Vegetable",B5="capsicum","Vegetable")`. Below the formula bar, a table titled "IFS Function" is displayed with the following data:

| Item name | When IFS used  | When IFS used with ELSE |
|-----------|--|-------------------------|
| Spinach   | <code>=IFS(B5="Apple","Fruit",B5="Banana","Fruit",B5="Spinach","Green Vegetable",B5="coffee","Beverage",B5="cabbage","Green Vegetable",B5="capsicum","Vegetable")</code> |                         |
| Capsicum  | Vegetable  | Vegetable               |
| Cabbage   | Green Vegetable  | Green Vegetable         |
| Coffee    | Beverage   | Beverage                |
| Apple     | Fruit  | Fruit                   |
| Banana    | Fruit  | Fruit                   |
| Pepper    | #N/A   | Misc                    |

Then, we will get the following result:

The screenshot shows the result of the IFS function. The formula bar contains the same formula as in the previous screenshot. The table below shows the results for each item:

| Item name | When IFS used   | When IFS used with ELSE |
|-----------|-----------------|-------------------------|
| Spinach   | Green Vegetable | Green Vegetable         |
| Capsicum  | Vegetable       | Vegetable               |
| Cabbage   | Green Vegetable | Green Vegetable         |
| Coffee    | Beverage        | Beverage                |
| Apple     | Fruit           | Fruit                   |
| Banana    | Fruit           | Fruit                   |
| Pepper    | #N/A            | Misc                    |

In this example, we have entered multiple logical tests in the IFS function. When a logical test evaluates to TRUE, the corresponding value will be returned. However, if none of the logical tests evaluate to TRUE, then IFS function would give #N/A error. This is what we got for Pepper in B8.

Using ELSE function, we can ensure that this #N/A error is removed. For this we would place a final logical test at the end of the formula that is TRUE and then place a value that shall be returned.

The screenshot shows the Excel formula bar with the following formula: `=IFS(B5="Apple","Fruit",B5="Banana","Fruit",B5="Spinach","Green Vegetable",B5="coffee","Beverage",B5="cabbage","Green Vegetable",B5="capsicum","Vegetable", TRUE, "Misc")`

| Item name | When IFS used   | When IFS used with ELSE   |
|-----------|-----------------|---|
| Spinach   | Green Vegetable | =IFS(B5="Apple","Fruit",B5="Banana","Fruit",B5="Spinach","Green Vegetable",B5="coffee","Beverage",B5="cabbage","Green Vegetable",B5="capsicum","Vegetable", TRUE, "Misc") |
| Capsicum  | Vegetable       | TRUE, "Misc")   |
| Cabbage   | Green Vegetable | Green Vegetable   |
| Coffee    | Beverage        | Beverage  |
| Apple     | Fruit           | Fruit   |
| Banana    | Fruit           | Fruit   |
| Pepper    | #N/A            | Misc  |

Here, we added TRUE, "Misc", this will ensure that Excel returns the value "Misc" in the scenario where none of the previous logical tests in the IFS function evaluates to TRUE.

Now the results would as be as shown below:

The screenshot shows the Excel formula bar with the same formula as above. The table below shows the updated results:

| Item name | When IFS used   | When IFS used with ELSE |
|-----------|-----------------|-------------------------|
| Spinach   | Green Vegetable | Green Vegetable         |
| Capsicum  | Vegetable       | Vegetable               |
| Cabbage   | Green Vegetable | Green Vegetable         |
| Coffee    | Beverage        | Beverage                |
| Apple     | Fruit           | Fruit                   |
| Banana    | Fruit           | Fruit                   |
| Pepper    | #N/A            | Misc                    |

## Example 2 – IFS vs nested IF function

Prior to the introduction of the IFS function, we used to use nested IF function. Let's see how the IFS function is superior to nested IF. Let's assume we wish to give discounts to customers based on their total purchases from us. So, a customer would get 5% discount for purchases more than \$100 but less than \$500, 10% discount if he made purchases of more than \$500 but less than \$750, 20% if he made purchases of more than \$750 but less than \$1,000 and 30% for all purchases above \$1,000.

Let's see the formula when we would have used nested IF function:

The screenshot shows the Excel formula bar with the following nested IF formula: `=IF(C5<500,5%,IF(C5<750,10%,IF(C5<1000,20%,IF(C5>1000,30%,0))))`. Below the formula bar, a table titled "IFS Function" is displayed:

| Name of Customer | Total purchases made by them | Discount using nested IF | Discount using IFS |
|------------------|------------------------------|--------------------------|--------------------|
|                  |                              |                          | 10%                |

Now, let's see how it becomes easy to write the formula for same results using IFS:

The screenshot shows the Excel formula bar with the following IFS formula: `=IFS(C5<100,0%,C5<500,5%,C5<750,10%,C5<1000,20%,C5>1000,30%,C5<100,0%)`. Below the formula bar, a table titled "IFS Function" is displayed:

| Name of Customer | Total purchases made by them | Discount using nested IF | Discount using IFS |
|------------------|------------------------------|--------------------------|--------------------|
|                  |                              |                          | 10%                |

Hence, IFS is easier as it allows using a single function to input a series of logical tests. It becomes cumbersome in the nested IF function, especially when we use a large number of logical tests.

**Few pointers on the IFS function**

1. #N/A Error occurs when no TRUE conditions are found by IFS function.
2. #VALUE! Error – We will get this error when the logical\_test argument resolves to a value other than TRUE or FALSE.
3. “You’ve entered too few arguments for this function” error message – This message appears when we provide a logical\_test argument without a corresponding value.

[Click here to download the sample Excel file](#)

**OR** Test multiple conditions with OR

logical1

logical2

...

### What is the OR Function?

The OR Function is categorized under **LOGICAL functions**. The function will determine if any of the conditions in the test is TRUE.

In **financial analysis**, the OR function can be useful in comparing two statements or two values. For example, A1 for either "a" or "b," use =OR(A1="a",A1="b"). The OR function can be used as the logical test inside the IF function to avoid extra nested IFs, and can be combined with the AND function.

### Formula

**=OR(logical1, [logical2], ...)**

The OR Function uses following arguments:

1. **Logical1** (required argument) – It is the first condition or logical value to evaluate.
2. **Logical2** (optional argument) – It is the second condition or logical value to evaluate.

### How to use the OR Function in Excel?

As a worksheet function, OR can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let us consider an example:

### Example 1

Let's see how we can test few conditions. Suppose we are given the following data:

|   | A | B                  | C             | D                        |
|---|---|--------------------|---------------|--------------------------|
| 1 |   |                    |               |                          |
| 2 |   | <b>OR Function</b> |               |                          |
| 3 |   |                    |               |                          |
| 4 |   | <b>Data 1</b>      | <b>Data 2</b> | <b>Formula</b>           |
| 5 |   | 6                  | 10            | OR(B4>0, B4<C4)          |
| 6 |   | 6                  | 10            | OR( B5>0, B5>C5, B5>12 ) |
| 7 |   | 6                  | 10            | OR( B6<0, B6>C6, C6>12 ) |

We get the results below:

|   | A | B                  | C             | D                        | E             |
|---|---|--------------------|---------------|--------------------------|---------------|
| 1 |   |                    |               |                          |               |
| 2 |   | <b>OR Function</b> |               |                          |               |
| 3 |   |                    |               |                          |               |
| 4 |   | <b>Data 1</b>      | <b>Data 2</b> | <b>Formula</b>           | <b>Result</b> |
| 5 |   | 6                  | 10            | OR(B4>0, B4<C4)          | TRUE          |
| 6 |   | 6                  | 10            | OR( B5>0, B5>C5, B5>12 ) | TRUE          |
| 7 |   | 6                  | 10            | OR( B6<0, B6>C6, C6>12 ) | FALSE         |

In the above examples:

1. The function in cell E4 evaluates to TRUE, as BOTH of the supplied conditions are TRUE;
2. The function in cell E5 evaluates to TRUE, as the first condition, B5>0 evaluates to TRUE;
3. The function in cell E6 evaluates to FALSE, as ALL of the supplied conditions are FALSE.

## Example 2

Let us now nest OR with IF. Suppose we are given student marks and we want the formula to return "PASS" if marks are above 50 in Economics or above 45 in Business Studies and "FAIL" for marks below it.

| Roll number | Marks in Eco | Marks in Business studies |
|-------------|--------------|---------------------------|
| 1           | 45           | 75                        |
| 2           | 55           | 65                        |
| 3           | 65           | 65                        |
| 4           | 50           | 85                        |
| 5           | 52           | 65                        |
| 6           | 48           | 35                        |

The formula to use is:

| Roll number | Marks in Eco | Marks in Business studies | Result |
|-------------|--------------|---------------------------|--------|
| 1           | 45           | 75                        | PASS   |
| 2           | 55           | 65                        | PASS   |
| 3           | 65           | 65                        | PASS   |
| 4           | 50           | 85                        | PASS   |
| 5           | 52           | 65                        | PASS   |
| 6           | 48           | 35                        | FAIL   |

We get the results below:

| Roll number | Marks in Eco | Marks in Business studies | Result |
|-------------|--------------|---------------------------|--------|
| 1           | 45           | 75                        | PASS   |
| 2           | 55           | 65                        | PASS   |
| 3           | 65           | 65                        | PASS   |
| 4           | 50           | 85                        | PASS   |
| 5           | 52           | 65                        | PASS   |
| 6           | 48           | 35                        | FAIL   |

Even if any one condition is satisfied, this function will return PASS.

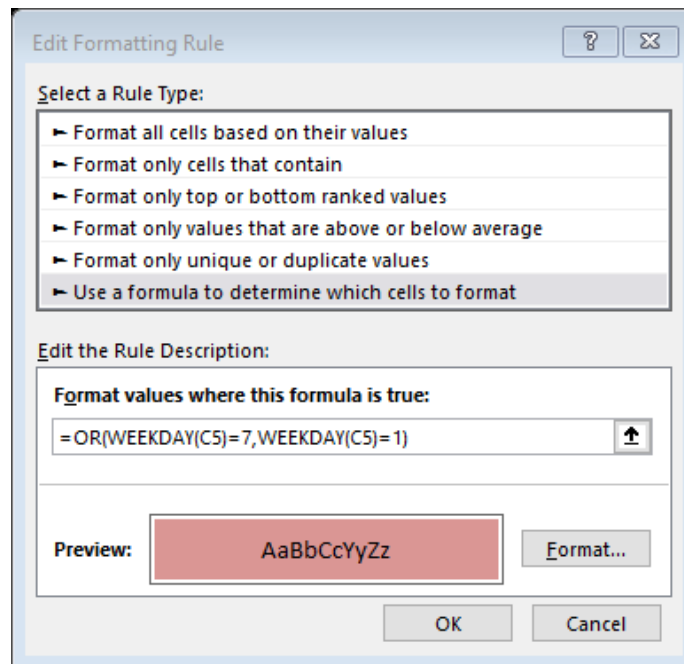


### Example 3

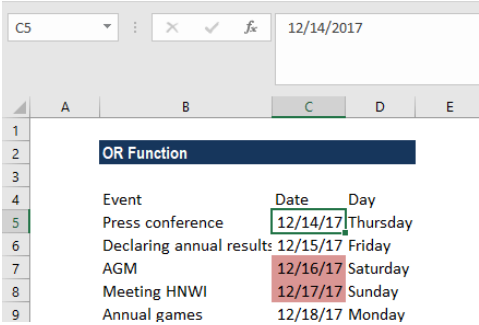
Suppose we want to highlight dates that occur on weekends. We can use the OR and WEEKDAY functions. Suppose we are given the data below:

|   | A | B                        | C        | D        |
|---|---|--------------------------|----------|----------|
| 1 |   |                          |          |          |
| 2 |   | OR Function              |          |          |
| 3 |   |                          |          |          |
| 4 |   | Event                    | Date     | Day      |
| 5 |   | Press conference         | 12/14/17 | Thursday |
| 6 |   | Declaring annual results | 12/15/17 | Friday   |
| 7 |   | AGM                      | 12/16/17 | Saturday |
| 8 |   | Meeting HNWI             | 12/17/17 | Sunday   |
| 9 |   | Annual games             | 12/18/17 | Monday   |

Using conditional formatting, we can highlight cells that fall on weekends. For that, select the cells B2:B6 and in Conditional Formatting, we will put the formula `=OR(WEEKDAY(B2)=7,WEEKDAY(B2)=1)`, as shown below:



We get the results below:



| Event                    | Date     | Day      |
|--------------------------|----------|----------|
| Press conference         | 12/14/17 | Thursday |
| Declaring annual results | 12/15/17 | Friday   |
| AGM                      | 12/16/17 | Saturday |
| Meeting HNW              | 12/17/17 | Sunday   |
| Annual games             | 12/18/17 | Monday   |

This formula uses the WEEKDAY function to test dates for either a Saturday or Sunday. When given a date, WEEKDAY returns a number 1-7, for each day of the week. In their standard configuration, Saturday = 7 and Sunday = 1. By using the OR function, we used WEEKDAY to test for either 1 or 7. If either is true, the formula will return TRUE and trigger the conditional formatting.

If we wish to highlight entire row, we need to apply the conditional formatting rule to all columns in the table and lock the date column. The formula to use will be =OR(WEEKDAY(\$B2)=7,WEEKDAY(\$B2)=1).

**Few things to remember about the OR Function:**

1. We can use the OR function to test multiple conditions at the same time – up to 255 conditions in total.
2. #VALUE! error – Occurs if any of the given logical\_test cannot be interpreted as numeric or logical values.
3. The OR function can also be used with AND function depending on the requirement.
4. If we enter OR as an array formula, we can test all values in a range against a condition. For example, the array formula will return TRUE if any cell in A1:A100 is greater than 15 =`{OR(A1:A100>15)}`.
5. The OR function will ignore the text values or empty cells provided in the arguments.

[Click here to download the sample Excel file](#)

## Lookup and Reference

**CHOOSE** Get a value from a list based on position `Index_num` `value2` `value2`

### What is the CHOOSE Function?

The CHOOSE function is categorized under [Lookup and Reference functions](#). It will return a value from an array corresponding to the index number provided. The function will return the *n-th* entry in a given list.

As a [financial analyst](#), the CHOOSE function is very useful when creating scenarios in [financial models](#). By using the CHOOSE formula, an analyst is able to select between 5 difference scenarios (for example) that can flow through the entire model. [Scenario analysis](#) is an important part of building a robust financial model.

|    | A   | B | C | D                  | E    | F    | G    | H    | I                      | J      | K      | L      | M      |
|----|---|---|---|--------------------|------|------|------|------|------------------------|--------|--------|--------|--------|
| 1  | © Corporate Finance Institute. All rights reserved. |   |   |                    |      |      |      |      |                        |        |        |        |        |
| 2  | <b>FINANCIAL STATEMENTS</b>                         |   |   | Historical Results |      |      |      |      | Forecast Period        |        |        |        |        |
| 3  |   |   |   | 2011               | 2012 | 2013 | 2014 | 2015 | 2016                   | 2017   | 2018   | 2019   | 2020   |
| 4  | Balance Sheet Check                                 |   |   | OK                 | OK   | OK   | OK   | OK   | OK                     | OK     | OK     | OK     | OK     |
| 7  | LIVE SCENARIO 1                                     |   |   |                    |      |      |      |      |                        |        |        |        |        |
| 8  | <b>SCENARIO 1</b>                                   |   |   |                    |      |      |      |      |                        |        |        |        |        |
| 9  | Revenue Growth (% Change)                           |   |   |                    |      |      |      |      | 10.0%                  | 10.0%  | 10.0%  | 10.0%  | 10.0%  |
| 10 | Cost of Goods Sold (% of Revenue)                   |   |   |                    |      |      |      |      | 42.0%                  | 47.0%  | 50.0%  | 36.0%  | 35.0%  |
| 11 | Salaries and Benefits (% of Revenue)                |   |   |                    |      |      |      |      | 17.0%                  | 17.0%  | 17.0%  | 17.0%  | 17.0%  |
| 12 | Rent and Overhead (\$000's)                         |   |   |                    |      |      |      |      | 15,000                 | 15,000 | 15,000 | 15,000 | 15,000 |
| 13 | Depreciation & Amortization (% of PP&E)             |   |   |                    |      |      |      |      | 35.0%                  | 35.0%  | 35.0%  | 35.0%  | 35.0%  |
| 22 | <b>SCENARIO 2</b>                                   |   |   |                    |      |      |      |      |                        |        |        |        |        |
| 24 | Revenue Growth (% Change)                           |   |   |                    |      |      |      |      | 5.0%                   | 4.5%   | 4.0%   | 3.5%   | 3.0%   |
| 25 | Cost of Goods Sold (% of Revenue)                   |   |   |                    |      |      |      |      | 37.0%                  | 37.0%  | 36.0%  | 36.0%  | 35.0%  |
| 26 | Salaries and Benefits (% of Revenue)                |   |   |                    |      |      |      |      | 17.0%                  | 17.0%  | 17.0%  | 17.0%  | 17.0%  |
| 27 | Rent and Overhead (\$000's)                         |   |   |                    |      |      |      |      | 10,000                 | 10,000 | 10,000 | 10,000 | 10,000 |
| 28 | Depreciation & Amortization (% of PP&E)             |   |   |                    |      |      |      |      | 40.0%                  | 40.0%  | 40.0%  | 40.0%  | 40.0%  |
| 37 | <b>LIVE SCENARIO</b>                                |   |   |                    |      |      |      |      |                        |        |        |        |        |
| 39 | Revenue Growth (% Change)                           |   |   |                    |      |      |      |      | =CHOOSE(\$J\$4,I9,I24) | 10.0%  | 10.0%  | 10.0%  | 10.0%  |
| 40 | Cost of Goods Sold (% of Revenue)                   |   |   |                    |      |      |      |      | 42.0%                  | 47.0%  | 50.0%  | 36.0%  | 35.0%  |
| 41 | Salaries and Benefits (% of Revenue)                |   |   |                    |      |      |      |      | 17.0%                  | 17.0%  | 17.0%  | 17.0%  | 17.0%  |
| 42 | Rent and Overhead (\$000's)                         |   |   |                    |      |      |      |      | 15,000                 | 15,000 | 15,000 | 15,000 | 15,000 |
| 43 | Depreciation & Amortization (% of PP&E)             |   |   |                    |      |      |      |      | 35.0%                  | 35.0%  | 35.0%  | 35.0%  | 35.0%  |

Learn more, in CFI's [scenario and sensitivity analysis course](#).

## Formula

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

The formula uses the following arguments:

1. Index\_num (required argument) – It is an integer that specifies which value argument is selected. Index\_num must be a number between 1 and 254, or a formula or reference to a cell containing a number between 1 and 254.
2. Value1, Value2 – Value1 is a required option but the rest are optional. It is a list of one or more values that we want to return a value from.

## Notes:

1. If index\_num is 1, CHOOSE returns value1; if it is 2, CHOOSE returns value2; and so on.
2. Value1, value2 must be entered as individual values (or references to individual cells containing values).
3. If the argument index\_num is a fraction, it is truncated to the lowest integer before being used.
4. If the argument index\_num is an array, every value is evaluated when CHOOSE is evaluated.
5. The value arguments can be range references, as well as single values.

## How to use the CHOOSE Function in Excel?

To understand the uses of the CHOOSE function, let's consider an example:

### Example 1

Suppose we are given the following dates:

|    | A | B                      | C |
|----|---|------------------------|---|
| 1  |   |                        |   |
| 2  |   | <b>CHOOSE Function</b> |   |
| 3  |   |                        |   |
| 4  |   | <b>Date</b>            |   |
| 5  |   | 31/01/2018             |   |
| 6  |   | 15/04/2018             |   |
| 7  |   | 15/07/2018             |   |
| 8  |   | 29/10/2018             |   |
| 9  |   | 01/01/2018             |   |
| 10 |   |                        |   |

We can calculate the fiscal quarter from the dates given above. The fiscal quarters start in a month other than January.

The formula to use is:

| IRR |   | =CHOOSE(MONTH(\$B5),1,1,1,2,2,2,3,3,3,4,4,4) |  |
|-----|---|--|--|
| A   | B | C  | D  |
| 1   |   |  |  |
| 2   |   | <b>CHOOSE Function</b>                       |  |
| 3   |   |  |  |
| 4   |   | <b>Date</b>                                  | <b>Jan Start</b> <b>Apr Start</b> <b>Jul Start</b> |
| 5   |   | 01/31/18                                     | =CHOOSE(MONTH(\$B5),1,1,1,2,2,2,3,3,3,4,4,4)       |

The formula returns a number from the array 1-4, which corresponds to a quarter system that begins in April and ends in March.

We get the results below:

| C5 |   | =CHOOSE(MONTH(\$B5),1,1,1,2,2,2,3,3,3,4,4,4) |  |
|----|---|--|--|
| A  | B | C  | D  |
| 1  |   |  |  |
| 2  |   | <b>CHOOSE Function</b>                       |  |
| 3  |   |  |  |
| 4  |   | <b>Date</b>                                  | <b>Jan Start</b> <b>Apr Start</b> <b>Jul Start</b> |
| 5  |   | 01/31/18                                     | 1  |
| 6  |   | 04/15/18                                     | 2  |
| 7  |   | 07/15/18                                     | 3  |
| 8  |   | 10/29/18                                     | 4  |
| 9  |   | 01/01/18                                     | 1  |
| 10 |   |  |  |

### Few things about the CHOOSE Function

1. VALUE! error – Occurs when:
  - The given index\_num is less than 1 or is greater than the given number of values.
  - The given index\_num argument is non-numeric.
2. #NAME? error – Occurs when the value arguments are text values that are not enclosed in quotes and are not valid cell references.

[Click here to download the sample Excel file.](#)

**HLOOKUP** Look up a value in a table by matching on the first row `value` `table` `row_index` `range_lookup`

### What is the HLOOKUP Function?

HLOOKUP stands for Horizontal Lookup and can be used to retrieve information from a table by searching a row for the matching data and outputting from the corresponding column. While VLOOKUP searches for the value in a column, HLOOKUP searches for the value in a row.

### Formula

**=HLOOKUP(value to look up, table area, row number)**

### How to use the HLOOKUP Function in Excel?

Let us consider the example below. The marks of four subjects for five students are as follows:

| Student name | A  | B  | C  | D  | E  |  |
|--------------|----|----|----|----|----|--|
| Accounting   | 75 | 65 | 70 | 60 | 59 |  |
| Economics    | 65 | 72 | 78 | 89 | 67 |  |
| Management   | 70 | 68 | 90 | 72 | 58 |  |
| Mathematics  | 80 | 90 | 75 | 65 | 87 |  |

Now if our objective is to fetch the marks of student D in Management, we can use HLOOKUP as follows:

|                                       | A   | B        | C        | D        | E        | F | G | H |
|---------------------------------------|---|----------|----------|----------|----------|---|---|---|
| <b>Student roll no</b>                | <b>A</b>  | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |   |   |   |
| Accounting                            | 75  | 65       | 70       | 60       | 59       |   |   |   |
| Economics                             | 65  | 72       | 78       | 89       | 67       |   |   |   |
| Management                            | 70  | 68       | 90       | 72       | 58       |   |   |   |
| Mathematics                           | 80  | 90       | 75       | 65       | 87       |   |   |   |
| <b>Fetch Marks of D in Management</b> | =HLOOKUP(<br>HLOOKUP( <b>lookup_value</b> , table_array, row_index_num, [range_lookup]) |          |          |          |          |   |   |   |

HLOOKUP function in Excel comes with the following arguments:

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



As you can see in the screenshot above, we need to give the lookup\_value first. Here, it would be student D as we need to find his marks in Management. Now, remember that lookup\_value can be a cell reference or a text string, or it can be a numerical value, too. In our example, it would be student name as shown below:

| A                      | B        | C        | D        | E        | F        | G | H |
|------------------------|----------|----------|----------|----------|----------|---|---|
| <b>Student roll no</b> | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |   |   |
| <b>Accounting</b>      | 75       | 65       | 70       | 60       | 59       |   |   |
| <b>Economics</b>       | 65       | 72       | 78       | 89       | 67       |   |   |
| <b>Management</b>      | 70       | 68       | 90       | 72       | 58       |   |   |
| <b>Mathematics</b>     | 80       | 90       | 75       | 65       | 87       |   |   |

|                                       |  |
|---------------------------------------|--|
| <b>Fetch Marks of D in Management</b> | =HLOOKUP("D"   |
|                                       | HLOOKUP( <b>lookup_value</b> , table_array, row_index_num, [range_lookup]) |

The next step would be to give the table array. Table array is nothing but rows of data in which the lookup value would be searched. Table array can be a regular range or a named range, or even an Excel table. Here we will give row A1:F5 as the reference.

| A                      | B        | C        | D        | E        | F        | G | H |
|------------------------|----------|----------|----------|----------|----------|---|---|
| <b>Student roll no</b> | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |   |   |
| <b>Accounting</b>      | 75       | 65       | 70       | 60       | 59       |   |   |
| <b>Economics</b>       | 65       | 72       | 78       | 89       | 67       |   |   |
| <b>Management</b>      | 70       | 68       | 90       | 72       | 58       |   |   |
| <b>Mathematics</b>     | 80       | 90       | 75       | 65       | 87       |   |   |

|                                       |   |
|---------------------------------------|---|
| <b>Fetch Marks of D in Management</b> | =HLOOKUP("D",A1:F5]   |
|                                       | HLOOKUP(lookup_value, <b>table_array</b> , row_index_num, [range_lookup]) |

Next, we would define 'row\_index\_num,' which is the row number in the table\_array from where the value would be returned. In this case, it would be 4 as we are fetching the value from the fourth row of the given table.

| A                      | B        | C        | D        | E        | F        | G | H |
|------------------------|----------|----------|----------|----------|----------|---|---|
| <b>Student roll no</b> | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> | <b>E</b> |   |   |
| <b>Accounting</b>      | 75       | 65       | 70       | 60       | 59       |   |   |
| <b>Economics</b>       | 65       | 72       | 78       | 89       | 67       |   |   |
| <b>Management</b>      | 70       | 68       | 90       | 72       | 58       |   |   |
| <b>Mathematics</b>     | 80       | 90       | 75       | 65       | 87       |   |   |

|                                       |   |
|---------------------------------------|---|
| <b>Fetch Marks of D in Management</b> | =HLOOKUP("D",A1:F5, 4]  |
|                                       | HLOOKUP(lookup_value, table_array, <b>row_index_num</b> , [range_lookup]) |

Suppose, if we require marks in Economics then we would put row\_index\_num as 3.

The next is range\_lookup. It makes HLOOKUP search for exact or approximate value. As we are looking out for exact value so it would be False.

|                 | A | B  | C  | D  | E  | F  | G | H | I | J | K | L | M | N | O |
|-----------------|---|----|----|----|----|----|---|---|---|---|---|---|---|---|---|
| Student roll no | A | B  | C  | D  | E  |    |   |   |   |   |   |   |   |   |   |
| Accounting      |   | 75 | 65 | 70 | 60 | 59 |   |   |   |   |   |   |   |   |   |
| Economics       |   | 65 | 72 | 78 | 89 | 67 |   |   |   |   |   |   |   |   |   |
| Management      |   | 70 | 68 | 90 | 72 | 58 |   |   |   |   |   |   |   |   |   |
| Mathematics     |   | 80 | 90 | 75 | 65 | 87 |   |   |   |   |   |   |   |   |   |

|                                |                        |
|--------------------------------|------------------------|
| Fetch Marks of D in Management | =HLOOKUP("D",A1:F5, 4, |
|--------------------------------|------------------------|

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

(-)- TRUE - Approximate match      Approximate match - the values in the first row of table\_array must be sorted in ascending order

(-)- FALSE - Exact match

The result would be 72.

|   |                                | A | B  | C  | D  | E  | F  | G |
|---|--------------------------------|---|----|----|----|----|----|---|
| 1 | Student roll no                | A | B  | C  | D  | E  |    |   |
| 2 | Accounting                     |   | 75 | 65 | 70 | 60 | 59 |   |
| 3 | Economics                      |   | 65 | 72 | 78 | 89 | 67 |   |
| 4 | Management                     |   | 70 | 68 | 90 | 72 | 58 |   |
| 5 | Mathematics                    |   | 80 | 90 | 75 | 65 | 87 |   |
| 6 |                                |   |    |    |    |    |    |   |
| 7 |                                |   |    |    |    |    |    |   |
| 8 | Fetch Marks of D in Management |   | 72 |    |    |    |    |   |

Here, HLOOKUP is searching for a particular value in the table and returning an exact or approximate value.

### Important points to keep in mind about HLOOKUP

1. It is a case-insensitive lookup. It will consider, for example, "TIM" and "tim" as same.
2. The 'Lookup\_value' should be the topmost row of the 'table\_array' when we are using HLOOKUP. If we need to look somewhere else, then we must use another Excel formula.
3. HLOOKUP supports wildcard characters such as '\*' or '?' in the 'lookup\_value' argument (only if 'lookup\_value' is text).

Let's understand this using an example.

Suppose we are given names of student and marks below:

| Student roll no | Amy | Brian | Cathy | Donald | Ela |
|-----------------|-----|-------|-------|--------|-----|
| Accounting      | 75  | 65    | 70    | 60     | 59  |
| Economics       | 65  | 72    | 78    | 89     | 67  |
| Management      | 70  | 68    | 90    | 72     | 58  |
| Mathematics     | 80  | 90    | 75    | 65     | 87  |

If we need to use the Horizontal Lookup formula to find the Math marks of a student whose name starts with a 'D,' the formula will be:

|                                |   |
|--------------------------------|---|
| Fetch Marks of D in Management | =HLOOKUP("D*",A1:F5, 4,FALSE)                                     |
|                                | HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup]) |

The wild character used is '\*'.

4. #N/A error – It would be returned by HLOOKUP if 'range\_lookup' is FALSE and HLOOKUP function is unable to find the 'lookup\_value' in the given range. We can embed the function in IFERROR and display our own message, for example: =IFERROR(HLOOKUP(A4, A1:I2, 2, FALSE), "No value found").
5. If the 'row\_index\_num' < 1, HLOOKUP would return #VALUE! error. If 'row\_index\_num' > number of columns in 'table\_array', then it would give #REF! error.
6. Remember HLOOKUP function in Excel can return only one value. This would be the first value n that would match the lookup value. What if there are few identical records in the table? In that scenario, it is advisable to remove them or create a Pivot table and group them. The array formula can then be used on Pivot table to extract all duplicate values that are present in the lookup range.

To learn more, launch our [free Excel crash course](#) now!

## HLOOKUP from another workbook or worksheet

It means giving an external reference to our HLOOKUP formula. Using the same table, the marks of students in subject Business Finance are given in sheet 2 as follows:

|    |                        |            |              |              |               |            |
|----|------------------------|------------|--------------|--------------|---------------|------------|
| 11 |                        |            |              |              |               |            |
| 12 | <b>Student roll no</b> | <b>Amy</b> | <b>Brian</b> | <b>Cathy</b> | <b>Donald</b> | <b>Ela</b> |
| 13 | <b>Accounting</b>      | 75         | 65           | 70           | 60            | 59         |
| 14 | <b>Economics</b>       | 65         | 72           | 78           | 89            | 67         |
| 15 | <b>Management</b>      | 70         | 68           | 90           | 72            | 58         |
| 16 | <b>Mathematics</b>     | 80         | 90           | 75           | 65            | 87         |
| 17 |                        |            |              |              |               |            |
| 18 |                        |            |              |              |               |            |
| 19 |                        |            |              |              |               |            |
| 20 |                        |            |              |              |               |            |
| 21 |                        |            |              |              |               |            |

|    |                         |            |              |              |               |            |
|----|-------------------------|------------|--------------|--------------|---------------|------------|
|    | <b>A</b>                | <b>B</b>   | <b>C</b>     | <b>D</b>     | <b>E</b>      | <b>F</b>   |
| 1  | <b>Student name</b>     | <b>Amy</b> | <b>Brian</b> | <b>Cathy</b> | <b>Donald</b> | <b>Ela</b> |
| 2  | <b>Business Finance</b> | 75         | 65           | 70           | 60            | 59         |
| 3  |                         |            |              |              |               |            |
| 4  |                         |            |              |              |               |            |
| 5  |                         |            |              |              |               |            |
| 6  |                         |            |              |              |               |            |
| 7  |                         |            |              |              |               |            |
| 8  |                         |            |              |              |               |            |
| 9  |                         |            |              |              |               |            |
| 10 |                         |            |              |              |               |            |
| 11 |                         |            |              |              |               |            |
| 12 |                         |            |              |              |               |            |
| 13 |                         |            |              |              |               |            |
| 14 |                         |            |              |              |               |            |
| 15 |                         |            |              |              |               |            |
| 16 |                         |            |              |              |               |            |
| 17 |                         |            |              |              |               |            |
| 18 |                         |            |              |              |               |            |
| 19 |                         |            |              |              |               |            |
| 20 |                         |            |              |              |               |            |
| 21 |                         |            |              |              |               |            |
| 22 |                         |            |              |              |               |            |
| 23 |                         |            |              |              |               |            |
| 24 |                         |            |              |              |               |            |

We will use following formula:

|                         |                                    |              |              |               |            |  |  |  |
|-------------------------|------------------------------------|--------------|--------------|---------------|------------|--|--|--|
| <b>Student name</b>     | <b>Amy</b>                         | <b>Brian</b> | <b>Cathy</b> | <b>Donald</b> | <b>Ela</b> |  |  |  |
| <b>Accounting</b>       | 75                                 | 65           | 70           | 60            | 59         |  |  |  |
| <b>Economics</b>        | 65                                 | 72           | 78           | 89            | 67         |  |  |  |
| <b>Management</b>       | 70                                 | 68           | 90           | 72            | 58         |  |  |  |
| <b>Mathematics</b>      | 80                                 | 90           | 75           | 65            | 87         |  |  |  |
| <b>Business Finance</b> | =HLOOKUP(B12,Sheet1!A1:B2,2,FALSE) |              |              |               |            |  |  |  |

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

Then we will drag it to the remaining cells.

| A                       | B          | C            | D            | E             | F          |
|-------------------------|------------|--------------|--------------|---------------|------------|
| <b>Student name</b>     | <b>Amy</b> | <b>Brian</b> | <b>Cathy</b> | <b>Donald</b> | <b>Ela</b> |
| <b>Accounting</b>       | 75         | 65           | 70           | 60            | 59         |
| <b>Economics</b>        | 65         | 72           | 78           | 89            | 67         |
| <b>Management</b>       | 70         | 68           | 90           | 72            | 58         |
| <b>Mathematics</b>      | 80         | 90           | 75           | 65            | 87         |
| <b>Business Finance</b> | 75         | 65           | 70           | 60            | 59         |

## Use of HLOOKUP to return multiple values from a single Horizontal LOOKUP

So far, we've used HLOOKUP for a single value. Now, let's use it to obtain multiple values.

| Student name     | Amy | Brian | Cathy | Donald | Ela |
|------------------|-----|-------|-------|--------|-----|
| Accounts         | 75  | 65    | 70    | 60     | 59  |
| Economics        | 65  | 72    | 78    | 89     | 67  |
| Management       | 70  | 68    | 90    | 72     | 58  |
| Mathematics      | 80  | 90    | 75    | 65     | 87  |
| Business Finance | 75  | 65    | 70    | 60     | 59  |

As shown in the table above, if I need to extract the marks of Cathy in all subjects, then I need to use the following formula:

```
=HLOOKUP("Cathy",B12:F17,{1,2,3,4,5},FALSE)
```

If you wish to get an array, you need to select the number of cells that are equal to the number of rows that you want HLOOKUP to return.

| Student name     | Amy | Brian | Cathy | Donald | Ela |
|------------------|-----|-------|-------|--------|-----|
| Accounting       | 75  | 65    | 70    | 60     | 59  |
| Economics        | 65  | 72    | 78    | 89     | 67  |
| Management       | 70  | 68    | 90    | 72     | 58  |
| Mathematics      | 80  | 90    | 75    | 65     | 87  |
| Business Finance | 75  | 65    | 70    | 60     | 59  |

Hlookup as an array formula 

```
=HLOOKUP("cathy",B12:F17,{1,2,3,4,5,6},FALSE)
```

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

After typing FALSE, we need to press Ctrl + Shift + Enter instead of the Enter key. Why do we need to do so?

Ctrl + Shift + Enter will enclose the HLOOKUP formula in curly brackets. As shown below, all cells will give the results in one go. We will be saved from typing the formula in each cell.

|    | A                                  | B          | C            | D            | E             | F          | G |
|----|------------------------------------|------------|--------------|--------------|---------------|------------|---|
| 10 |                                    |            |              |              |               |            |   |
| 11 |                                    |            |              |              |               |            |   |
| 12 | <b>Student name</b>                | <b>Amy</b> | <b>Brian</b> | <b>Cathy</b> | <b>Donald</b> | <b>Ela</b> |   |
| 13 | <b>Accounting</b>                  | 75         | 65           | 70           | 60            | 59         |   |
| 14 | <b>Economics</b>                   | 65         | 72           | 78           | 89            | 67         |   |
| 15 | <b>Management</b>                  | 70         | 68           | 90           | 72            | 58         |   |
| 16 | <b>Mathematics</b>                 | 80         | 90           | 75           | 65            | 87         |   |
| 17 | <b>Business Finance</b>            | 75         | 65           | 70           | 60            | 59         |   |
| 18 |                                    |            |              |              |               |            |   |
| 19 |                                    |            |              |              |               |            |   |
| 20 | <b>Hlookup as an array formula</b> | Cathy      | 70           | 78           | 90            | 75         |   |
| 21 |                                    |            |              |              |               |            |   |
| 22 |                                    |            |              |              |               |            |   |

**INDEX** Get a value in a list or a table based on location

array

row\_num

col\_num

area\_num

## What is the INDEX Function?

The INDEX function is categorized under **Lookup and Reference functions**. The function will return the value at a given position in a range or array. The INDEX function is often used with the **MATCH function**. We can say it is an alternative way to do VLOOKUP.

As a **financial analyst**, INDEX can be used in other forms of analysis besides looking up a value in a list or table. In financial analysis, we can use it along with other functions, for lookup and return the sum of a column.

There are two formats for the INDEX function:

1. Array format
2. Reference format

## The Array format of the INDEX Function

The array format is used when we wish to return the value of a specified cell or array of cells.

### Formula

**=INDEX(array, row\_num, [col\_num])**

The function uses the following arguments:

1. **Array** (required argument) – It is the specified array or range of cells.
2. **Row\_num** (required argument) – It denotes the row number of the specified array. When the argument is set to zero or blank, it will default to all rows in the array provided.
3. **Col\_num** (optional argument) – It denotes the column number of the specified array. When this argument is set to zero or blank, it will default to all rows in the array provided.



## The Reference format of the INDEX Function

The reference format is used when we wish to return the reference of the cell at the intersection row\_num and col\_num.

### Formula

**=INDEX(reference, row\_num, [column\_num], [area\_num])**

The function uses the following arguments:

1. **Reference** (required argument) – It is a reference to one or more cells. If we input multiple areas directly into the function, individual areas should be separated by commas and surrounded by brackets. Such as (A1:B2, C3:D4), etc.
2. **Row\_num** (required argument) – It denotes the row number of a specified area. When the argument is set to zero or blank, it will default to all rows in the array provided.
3. **Col\_num** (optional argument) – It denotes the column number of the specified array. When the argument is set to zero or blank, it will default to all rows in the array provided.
4. **Area\_num** (optional argument) – If the reference is supplied as multiple ranges, area\_num indicates which range to use. Areas are numbered by the order they are specified.

If the area\_num argument is omitted, it defaults to the value 1 (i.e., the reference is taken from the first area in the supplied range).

### How to use the INDEX Function in Excel?

To understand the uses of the function, let us consider a few examples:

### Example 1

We are given the following data and we wish to match the location of a value.

|   | A | B                      | C               | D                       | E            |
|---|---|------------------------|-----------------|-------------------------|--------------|
| 1 |   |                        |                 |                         |              |
| 2 |   | <b>INDEX Function</b>  |                 |                         |              |
| 3 |   |                        |                 |                         |              |
| 4 |   | <b>Name of Athlete</b> | <b>Position</b> | <b>Distance covered</b> | <b>Speed</b> |
| 5 |   | Jimmy                  | 1               | 15                      | 3km/hr       |
| 6 |   | Lily                   | 2               | 14                      | 2.8km/hr     |
| 7 |   | William                | 3               | 13                      | 2.7 km/hr    |
| 8 |   | Rashmi                 | 2               | 14                      | 2.8km/hr     |
| 9 |   | Samar                  | 4               | 12                      | 2.7 km/hr    |

In the table above, we wish to see the distance covered by William. The formula to use will be:

|    | A | B                           | C               | D                       | E            | F |
|----|---|-----------------------------|-----------------|-------------------------|--------------|---|
| 1  |   |                             |                 |                         |              |   |
| 2  |   | <b>INDEX Function</b>       |                 |                         |              |   |
| 3  |   |                             |                 |                         |              |   |
| 4  |   | <b>Name of Athlete</b>      | <b>Position</b> | <b>Distance covered</b> | <b>Speed</b> |   |
| 5  |   | Jimmy                       | 1               | 15                      | 3km/hr       |   |
| 6  |   | Lily                        | 2               | 14                      | 2.8km/hr     |   |
| 7  |   | William                     | 3               | 13                      | 2.7 km/hr    |   |
| 8  |   | Rashmi                      | 2               | 14                      | 2.8km/hr     |   |
| 9  |   | Samar                       | 4               | 12                      | 2.7 km/hr    |   |
| 10 |   |                             |                 |                         |              |   |
| 11 |   | Distance covered by William |                 | =INDEX(B5:E9,3,3)       |              |   |

We get the result below:

|    | A | B                           | C               | D                       | E            | F |
|----|---|-----------------------------|-----------------|-------------------------|--------------|---|
| 1  |   |                             |                 |                         |              |   |
| 2  |   | <b>INDEX Function</b>       |                 |                         |              |   |
| 3  |   |                             |                 |                         |              |   |
| 4  |   | <b>Name of Athlete</b>      | <b>Position</b> | <b>Distance covered</b> | <b>Speed</b> |   |
| 5  |   | Jimmy                       | 1               | 15                      | 3km/hr       |   |
| 6  |   | Lily                        | 2               | 14                      | 2.8km/hr     |   |
| 7  |   | William                     | 3               | 13                      | 2.7 km/hr    |   |
| 8  |   | Rashmi                      | 2               | 14                      | 2.8km/hr     |   |
| 9  |   | Samar                       | 4               | 12                      | 2.7 km/hr    |   |
| 10 |   |                             |                 |                         |              |   |
| 11 |   | Distance covered by William |                 | 13                      |              |   |

## Example 2

Now let's see how to use the MATCH and INDEX functions at the same time. Suppose we are given the following data:

|    | A | B              | C              | D                          | E                          | F |
|----|---|----------------|----------------|----------------------------|----------------------------|---|
| 1  |   |                |                |                            |                            |   |
| 2  |   | INDEX Function |                |                            |                            |   |
| 3  |   |                |                |                            |                            |   |
| 4  |   | <b>Rank</b>    | <b>Country</b> | <b>Starting a Business</b> | <b>Getting Electricity</b> |   |
| 5  |   | 1              | New Zeland     | 1                          | 37                         |   |
| 6  |   | 2              | Singapore      | 6                          | 12                         |   |
| 7  |   | 3              | Denmark        | 34                         | 16                         |   |
| 8  |   | 4              | Korea Rep.     | 9                          | 2                          |   |
| 9  |   | 5              | Hong Kong      | 3                          | 4                          |   |
| 10 |   | 6              | US             | 49                         | 49                         |   |
| 11 |   | 7              | UK             | 14                         | 9                          |   |
| 12 |   | 8              | Norway         | 19                         | 23                         |   |
| 13 |   | 9              | Georgia        | 4                          | 30                         |   |
| 14 |   | 10             | Sweden         | 13                         | 6                          |   |

Suppose we wish to find out Georgia's rank in the Ease of Doing Business category. We will use the following formula:

|    | A | B              | C              | D                          | E                          | F | G | H |
|----|---|----------------|----------------|----------------------------|----------------------------|---|---|---|
| 2  |   | INDEX Function |                |                            |                            |   |   |   |
| 3  |   |                |                |                            |                            |   |   |   |
| 4  |   | <b>Rank</b>    | <b>Country</b> | <b>Starting a Business</b> | <b>Getting Electricity</b> |   |   |   |
| 5  |   | 1              | New Zeland     | 1                          | 37                         |   |   |   |
| 6  |   | 2              | Singapore      | 6                          | 12                         |   |   |   |
| 7  |   | 3              | Denmark        | 34                         | 16                         |   |   |   |
| 8  |   | 4              | Korea Rep.     | 9                          | 2                          |   |   |   |
| 9  |   | 5              | Hong Kong      | 3                          | 4                          |   |   |   |
| 10 |   | 6              | US             | 49                         | 49                         |   |   |   |
| 11 |   | 7              | UK             | 14                         | 9                          |   |   |   |
| 12 |   | 8              | Norway         | 19                         | 23                         |   |   |   |
| 13 |   | 9              | Georgia        | 4                          | 30                         |   |   |   |
| 14 |   | 10             | Sweden         | 13                         | 6                          |   |   |   |
| 15 |   |                |                |                            |                            |   |   |   |
| 16 |   |                |                |                            |                            |   |   |   |
| 17 |   |                |                |                            |                            |   |   |   |

Formula bar: `=INDEX($D$5:$D$14,MATCH("Georgia",$C$5:$C$14,0))`

Formula bar: `=INDEX($D$5:$D$14,MATCH("Georgia",$C$5:$C$14,0))`

Here, the MATCH function will look up for Georgia and return number 10 as Georgia is 10 on the list. The INDEX function takes "10" in the second parameter (row\_num), which indicates which row we wish to return a value from and turns into a simple `=INDEX($C$2:$C$11,3)`.

We get the result below:

The screenshot shows an Excel spreadsheet with the following data:

| Rank | Country     | Starting a Business | Getting Electricity |
|------|-------------|---------------------|---------------------|
| 1    | New Zealand | 1                   | 37                  |
| 2    | Singapore   | 6                   | 12                  |
| 3    | Denmark     | 34                  | 16                  |
| 4    | Korea Rep.  | 9                   | 2                   |
| 5    | Hong Kong   | 3                   | 4                   |
| 6    | US          | 49                  | 49                  |
| 7    | UK          | 14                  | 9                   |
| 8    | Norway      | 19                  | 23                  |
| 9    | Georgia     | 4                   | 30                  |
| 10   | Sweden      | 13                  | 6                   |

The formula bar shows: `=INDEX($D$5:$D$14,MATCH("Georgia",$C$5:$C$14,0))`

The result of the formula in cell D17 is 4.

### Things to remember about the INDEX Function

1. #VALUE! error – Occurs when any of the given row\_num, col\_num or area\_num arguments are non-numeric.
2. #REF! error – Occurs due to either of the following reasons:
  - The given row\_num argument is greater than the number of rows in the given range;
  - The given [col\_num] argument is greater than the number of columns in the range provided; or
  - The given [area\_num] argument is more than the number of areas in the supplied range.
3. VLOOKUP vs. INDEX function
  - Excel VLOOKUP is unable to look to its left, meaning that our lookup value should always reside in the left-most column of the lookup range. This is not the case with the INDEX and MATCH functions.
  - VLOOKUP formulas get broken or return incorrect results when a new column is deleted from or added to a lookup table. With INDEX and MATCH, we can delete or insert new columns in your lookup table without distorting the results.

[Click here to download the sample Excel file](#)

**MATCH** Get the position of an item in an `lookup_value` `lookup_array` `match_type` array

### What is the MATCH Function?

The MATCH function is categorized under **Lookup and Reference functions**. It looks up a value in an array and returns the position of the value within the array. For example, if we wish to match the value 5 in the range A1:A4, which contains values 1,5,3,8, the function will return 2 as 5 is the second item in the range.

In **financial analysis**, we can use the MATCH function along with other functions to look up and return the sum of values in a column. It is commonly used with the INDEX function. Learn how to combine **INDEX MATCH** as a powerful lookup combination.

### Formula

**=MATCH(lookup\_value, lookup\_array, [match\_type])**

The MATCH formula uses the following arguments:

1. **Lookup\_value** (required argument) – It is the value that we want to look up.
2. **Lookup\_array** (required argument) – It is the data array that is to be searched.
3. **Match\_type** (optional argument) – It can be set to 1, 0 or -1 to return results as given below:

| Match_type   | Behavior   |
|--------------|--|
| 1 or omitted | When the function cannot find an exact match, it will return the position of the closest match below the lookup_value. (If this option is used, the lookup_array must be in ascending order).  |
| 0            | When the function cannot find an exact match, it will return an error. (If this option is used, the lookup_array does not need to be ordered).   |
| -1           | When the function cannot find an exact match, it will return the position of the closest match above the lookup_value. (If this option is used, the lookup_array must be in descending order). |

### How to use the MATCH Function in Excel?

To understand the uses of the MATCH function, let's consider a few examples:

### Example 1

Suppose we are given the following data:

|    | A                     | B                 | C |
|----|-----------------------|-------------------|---|
| 1  |                       |                   |   |
| 2  | <b>MATCH Function</b> |                   |   |
| 3  |                       |                   |   |
| 4  |                       | <b>Vegetables</b> |   |
| 5  |                       |                   |   |
| 6  |                       | Spinach           |   |
| 7  |                       | Cucumber          |   |
| 8  |                       | Capsicum          |   |
| 9  |                       | Broccoli          |   |
| 10 |                       | Fenugreek         |   |
| 11 |                       | Gourd             |   |

We can use the following formula to find the value for Cucumber.

| IRR |                       |                   |   | =MATCH(D6,B6:B11,0) |                     |   |  |
|-----|-----------------------|-------------------|---|---------------------|---------------------|---|--|
| A   | B                     | C                 | D | E                   | F                   | G |  |
| 1   |                       |                   |   |                     |                     |   |  |
| 2   | <b>MATCH Function</b> |                   |   |                     |                     |   |  |
| 3   |                       |                   |   |                     |                     |   |  |
| 4   |                       | <b>Vegetables</b> |   |                     |                     |   |  |
| 5   |                       |                   |   |                     |                     |   |  |
| 6   |                       | Spinach           |   | Value               | Result              |   |  |
| 7   |                       | Cucumber          |   | Cucumber            | =MATCH(D6,B6:B11,0) |   |  |
| 8   |                       | Capsicum          |   |                     |                     |   |  |
| 9   |                       | Broccoli          |   |                     |                     |   |  |
| 10  |                       | Fenugreek         |   |                     |                     |   |  |
| 11  |                       | Gourd             |   |                     |                     |   |  |
| 12  |                       |                   |   |                     |                     |   |  |

We get the result below:

| E6 |                       |                   |   | =MATCH(D6,B6:B11,0) |        |  |  |
|----|-----------------------|-------------------|---|---------------------|--------|--|--|
| A  | B                     | C                 | D | E                   | F      |  |  |
| 1  |                       |                   |   |                     |        |  |  |
| 2  | <b>MATCH Function</b> |                   |   |                     |        |  |  |
| 3  |                       |                   |   |                     |        |  |  |
| 4  |                       | <b>Vegetables</b> |   |                     |        |  |  |
| 5  |                       |                   |   | Value               | Result |  |  |
| 6  |                       | Spinach           |   | Cucumber            | 2      |  |  |
| 7  |                       | Cucumber          |   |                     |        |  |  |
| 8  |                       | Capsicum          |   |                     |        |  |  |
| 9  |                       | Broccoli          |   |                     |        |  |  |
| 10 |                       | Fenugreek         |   |                     |        |  |  |
| 11 |                       | Gourd             |   |                     |        |  |  |
| 12 |                       |                   |   |                     |        |  |  |

## Example 2

Suppose we are given the following data:

|    | A | B              | C    | D     | E        | F |
|----|---|----------------|------|-------|----------|---|
| 1  |   |                |      |       |          |   |
| 2  |   | MATCH Function |      |       |          |   |
| 3  |   |                |      |       |          |   |
| 4  |   | Item           | Size | Color | Quantity |   |
| 5  |   | Jeans          | S    | Blue  | 100      |   |
| 6  |   | Trousers       | S    | Brown | 125      |   |
| 7  |   | Trousers       | L    | Black | 145      |   |
| 8  |   | Trousers       | M    | Blue  | 245      |   |
| 9  |   | Jeans          | M    | Blue  | 500      |   |
| 10 |   | Jeans          | L    | Blue  | 150      |   |
| 11 |   | Capris         | S    | Black | 180      |   |
| 12 |   | Shorts         | L    | Red   | 190      |   |
| 13 |   | Shorts         | S    | Blue  | 215      |   |

Suppose we wish to find out the number of Trousers of a specific color. Here, we will use the INDEX or MATCH functions. The array formula to use is:

IRR : X ✓ fx =INDEX(E5:E13,MATCH(1,(I7=B5:B13)\*(I8=C5:C13)\*(I9=D5:D13),0))

|    | A | B              | C    | D     | E        | F | G | H | I | J | K | L | M | N | O |
|----|---|----------------|------|-------|----------|---|---|---|---|---|---|---|---|---|---|
| 1  |   |                |      |       |          |   |   |   |   |   |   |   |   |   |   |
| 2  |   | MATCH Function |      |       |          |   |   |   |   |   |   |   |   |   |   |
| 3  |   |                |      |       |          |   |   |   |   |   |   |   |   |   |   |
| 4  |   | Item           | Size | Color | Quantity |   |   |   |   |   |   |   |   |   |   |
| 5  |   | Jeans          | S    | Blue  | 100      |   |   |   |   |   |   |   |   |   |   |
| 6  |   | Trousers       | S    | Brown | 125      |   |   |   |   |   |   |   |   |   |   |
| 7  |   | Trousers       | L    | Black | 145      |   |   |   |   |   |   |   |   |   |   |
| 8  |   | Trousers       | M    | Blue  | 245      |   |   |   |   |   |   |   |   |   |   |
| 9  |   | Jeans          | M    | Blue  | 500      |   |   |   |   |   |   |   |   |   |   |
| 10 |   | Jeans          | L    | Blue  | 150      |   |   |   |   |   |   |   |   |   |   |
| 11 |   | Capris         | S    | Black | 180      |   |   |   |   |   |   |   |   |   |   |
| 12 |   | Shorts         | L    | Red   | 190      |   |   |   |   |   |   |   |   |   |   |
| 13 |   | Shorts         | S    | Blue  | 215      |   |   |   |   |   |   |   |   |   |   |
| 14 |   |                |      |       |          |   |   |   |   |   |   |   |   |   |   |

Item: Trousers  
Size: L  
Color: Black  
Quantity: =INDEX(E5:E13,MATCH(1,(I7=B5:B13)\*(I8=C5:C13)\*(I9=D5:D13),0))

We need to create an array using CTRL + SHIFT + ENTER. We get the result below:

I10 : X ✓ fx {=INDEX(E5:E13,MATCH(1,(I7=B5:B13)\*(I8=C5:C13)\*(I9=D5:D13),0))}

|    | A | B              | C    | D     | E        | F | G | H | I | J | K |  |
|----|---|----------------|------|-------|----------|---|---|---|---|---|---|--|
| 1  |   |                |      |       |          |   |   |   |   |   |   |  |
| 2  |   | MATCH Function |      |       |          |   |   |   |   |   |   |  |
| 3  |   |                |      |       |          |   |   |   |   |   |   |  |
| 4  |   | Item           | Size | Color | Quantity |   |   |   |   |   |   |  |
| 5  |   | Jeans          | S    | Blue  | 100      |   |   |   |   |   |   |  |
| 6  |   | Trousers       | S    | Brown | 125      |   |   |   |   |   |   |  |
| 7  |   | Trousers       | L    | Black | 145      |   |   |   |   |   |   |  |
| 8  |   | Trousers       | M    | Blue  | 245      |   |   |   |   |   |   |  |
| 9  |   | Jeans          | M    | Blue  | 500      |   |   |   |   |   |   |  |
| 10 |   | Jeans          | L    | Blue  | 150      |   |   |   |   |   |   |  |
| 11 |   | Capris         | S    | Black | 180      |   |   |   |   |   |   |  |
| 12 |   | Shorts         | L    | Red   | 190      |   |   |   |   |   |   |  |
| 13 |   | Shorts         | S    | Blue  | 215      |   |   |   |   |   |   |  |

Item: Trousers  
Size: L  
Color: Black  
Quantity: 145

**Things to remember about the MATCH Function**

1. The MATCH function does not distinguish between uppercase and lowercase letters when matching text values.
2. N/A! error – Occurs if the match function fails to find a match for the lookup\_value.
3. The function supports approximate and exact matching and wildcards (\* or ?) for partial matches.

[Click here to download the sample Excel file.](#)



**OFFSET** Create a reference offset from given starting point

reference row cols height width

### What is the OFFSET Function?

The OFFSET function is categorized under [Lookup and Reference functions](#). OFFSET will return a range of cells. That is, it will return a specified number of rows and columns from an initial range that was specified.

In [financial analysis](#), we often use Pivots Tables and Charts. The OFFSET function can be used to build a dynamic named range for pivot tables or charts to make sure that the source data is always up to date.

### Formula

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

The OFFSET function uses the following arguments:

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

### How to use the OFFSET Function in Excel?

As a worksheet function, the OFFSET function can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let us consider a few examples:

### Example 1

Let's say we are given the weekly earnings for 5 weeks below:

|    | A | B                      | C | D | E | F | G | H |
|----|---|------------------------|---|---|---|---|---|---|
| 1  |   |                        |   |   |   |   |   |   |
| 2  |   | <b>OFFSET Function</b> |   |   |   |   |   |   |
| 3  |   |                        |   |   |   |   |   |   |
| 4  |   |                        |   |   |   |   |   |   |
| 5  |   |                        |   |   |   |   |   |   |
| 6  |   |                        |   |   |   |   |   |   |
| 7  |   |                        |   |   |   |   |   |   |
| 8  |   |                        |   |   |   |   |   |   |
| 9  |   |                        |   |   |   |   |   |   |
| 10 |   |                        |   |   |   |   |   |   |
| 11 |   |                        |   |   |   |   |   |   |

| Day/Week  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
|-----------|--------|--------|--------|--------|--------|
| Monday    | 4,500  | 5,000  | 5,000  | 5,000  | 5,000  |
| Tuesday   | 3,500  | 2,000  | 3,000  | 3,000  | 3,000  |
| Wednesday | 2,500  | 3,200  | 3,500  | 2,500  | 400    |
| Thursday  | 4,200  | 4,800  | 4,500  | 1,500  | 2,500  |
| Friday    | 5,800  | 3,200  | 4,400  | 1,500  | 500    |

Now, if we insert the formula =OFFSET(A3,3,1) in cell B1, it will give us the value 2,500, which is 3 rows down on column right, as shown below:

|    | A | B                      | C    | D | E | F | G | H |
|----|---|------------------------|------|---|---|---|---|---|
| 1  |   |                        |      |   |   |   |   |   |
| 2  |   | <b>OFFSET Function</b> |      |   |   |   |   |   |
| 3  |   |                        |      |   |   |   |   |   |
| 4  |   |                        |      |   |   |   |   |   |
| 5  |   |                        |      |   |   |   |   |   |
| 6  |   |                        |      |   |   |   |   |   |
| 7  |   |                        |      |   |   |   |   |   |
| 8  |   |                        |      |   |   |   |   |   |
| 9  |   |                        |      |   |   |   |   |   |
| 10 |   |                        |      |   |   |   |   |   |
| 11 |   |                        |      |   |   |   |   |   |
| 12 |   |                        |      |   |   |   |   |   |
| 13 |   |                        | 2500 |   |   |   |   |   |

In the above example, as the height and width of the returned range are same as the reference range, we omitted the reference range.

## Example 2

Continuing with the same example, suppose we want the earnings for Thursday for all the weeks. In this scenario, we will use the formula `{=OFFSET(B7,3,1,1,5)}`.

We get the results below:

| Day/Week  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
|-----------|--------|--------|--------|--------|--------|
| Monday    | 4,500  | 5,000  | 5,000  | 5,000  | 5,000  |
| Tuesday   | 3,500  | 2,000  | 3,000  | 3,000  | 3,000  |
| Wednesday | 2,500  | 3,200  | 3,500  | 2,500  | 400    |
| Thursday  | 4,200  | 4,800  | 4,500  | 1,500  | 2,500  |
| Friday    | 5,800  | 3,200  | 4,400  | 1,500  | 500    |

In the example above:

1. As the results of the OFFSET function are to occupy more than one cell, it is necessary to enter the function as an array formula. It can be seen by the curly braces that surround the formula in the Formula bar.
2. In the given formula, as the width of the returned range is more than the width of the reference range, we specified the height and width arguments.

### Example 3

Continuing with the same example, suppose we want the sum total of the earnings for Week 3. We use the formula `=SUM(OFFSET(G6,1,-2,5))`.

The screenshot shows the Excel interface with the formula bar containing `=SUM(OFFSET(G6,1,-2,5))`. The spreadsheet below has the following data:

| Day/Week  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
|-----------|--------|--------|--------|--------|--------|
| Monday    | 4,500  | 5,000  | 5,000  | 5,000  | 5,000  |
| Tuesday   | 3,500  | 2,000  | 3,000  | 3,000  | 3,000  |
| Wednesday | 2,500  | 3,200  | 3,500  | 2,500  | 400    |
| Thursday  | 4,200  | 4,800  | 4,500  | 1,500  | 2,500  |
| Friday    | 5,800  | 3,200  | 4,400  | 1,500  | 500    |

We get the results below:

The screenshot shows the Excel interface with the formula bar containing `=SUM(OFFSET(G6,1,-2,5))`. The spreadsheet below shows the result of the formula in cell C4:

| Day/Week  | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 |
|-----------|--------|--------|--------|--------|--------|
| Monday    | 4,500  | 5,000  | 5,000  | 5,000  | 5,000  |
| Tuesday   | 3,500  | 2,000  | 3,000  | 3,000  | 3,000  |
| Wednesday | 2,500  | 3,200  | 3,500  | 2,500  | 400    |
| Thursday  | 4,200  | 4,800  | 4,500  | 1,500  | 2,500  |
| Friday    | 5,800  | 3,200  | 4,400  | 1,500  | 500    |

In the example above:

1. The array of values returned by the OFFSET function is directly entered into the SUM function, which returns a single value. Hence, we do not need it to enter as an array formula.
2. The height of the offset range is greater than the height of the reference range and so the [height] argument is entered as the value 5.
3. The width of the offset range is the same as the width of the reference range and so the [width] argument was omitted from the function.

**Few notes about the OFFSET Function:**

1. #REF! error – Occurs when the range resulting from the requested offset is invalid. For example, it extends beyond the edge of the worksheet.
2. #VALUE! error – Occurs if any of the supplied rows, cols, [height] or [width] arguments is non-numeric.

[Click here to download the sample Excel file](#)

**VLOOKUP** Lookup a value in a table by matching on the first column

value

table

col\_index

range\_lookup

### What is VLOOKUP in Excel?

The VLOOKUP function in Excel is a tool for looking up a piece of information in a table or dataset and extracting some corresponding data/information. In simple terms, the VLOOKUP function says the following to Excel, “look for this piece of information (i.e. bananas), in this dataset (a table), and tell me some corresponding information about it (i.e. the price of bananas)”.

Learn how to do this step by step in our [Free Excel Crash Course!](#)

### VLOOKUP Formula

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

To translate this to English the formula is saying, “look for this piece of information, in the following area, and give me some corresponding data in another column”.

The VLOOKUP function uses the following arguments:

1. **Lookup\_value** (required argument) – It is the value that we want to look up for in the first column of a table.
2. **Table\_array** (required argument) – It is the data array that is to be searched. The VLOOKUP function searches in the left-most column of this array.
3. **Col\_index\_num** (required argument) – An integer, specifying the column number of the supplied table\_array, that you want to return a value from.
4. **Range\_lookup** (optional argument) – It defines what this function should return in the event that it does not find an exact match to the lookup\_value. The argument can be set to TRUE or FALSE, which means:
  - TRUE – Approximate match, that is, if an exact match is not found, use the closest match below the lookup\_value.
  - FALSE – Exact match, that is, if an exact match not found then it will return an error.

## How to use VLOOKUP in Excel

### Step 1: Organize the data

The first step to effectively using the VLOOKUP function is to make sure your data is well organized and suitable for using the function.

VLOOKUP works in a left to right order, so you need to ensure that the information you want to look up is to the left of the corresponding data you want to extract.

For example:

| Good Table |           |               | Bad Table |               |           |
|------------|-----------|---------------|-----------|---------------|-----------|
| Fruit      | In Stock? | Price (\$/lb) | In Stock? | Price (\$/lb) | Fruit     |
| Grapes     | Yes       | 7.25          | Yes       | 7.25          | Grapes    |
| Mangos     | Yes       | 12.32         | Yes       | 12.32         | Mangos    |
| Bananas    | Yes       | 5.42          | Yes       | 5.42          | Bananas   |
| Pineapple  | No        | 7.18          | No        | 7.18          | Pineapple |
| Oranges    | Yes       | 8.99          | Yes       | 8.99          | Oranges   |

| Fruit   | Price | Fruit   | Price |
|---------|-------|---------|-------|
| Bananas | 5.42  | Bananas | #N/A  |

In the above VLOOKUP example, you will see that the “good table” can easily run the function to lookup “Bananas” and return their price, since Bananas are located in the leftmost column. In the “bad table” example you’ll see there is an error message, as the columns are not in the right order.

This is one of the major drawbacks of VLOOKUP, and for this reason, it’s highly recommended to use INDEX MATCH MATCH instead of VLOOKUP.

## Step 2: Tell the function what to lookup

In this step we tell Excel what to look for. We start typing the formula “=VLOOKUP(“ and then select the cell that contains the information we want to lookup. In this case, it’s the cell that contains “Bananas”.

The screenshot shows an Excel spreadsheet with the following data:

| Fruit     | In Stock? | Price (\$/lb) |
|-----------|-----------|---------------|
| Grapes    | Yes       | 7.25          |
| Mangos    | Yes       | 12.32         |
| Bananas   | Yes       | 5.42          |
| Pineapple | No        | 7.18          |
| Oranges   | Yes       | 8.99          |

| Fruit   | Price                       |
|---------|-----------------------------|
| Bananas | =VLOOKUP(A11,A4:C8,3,FALSE) |

1. Look for this item



### Step 3: Tell the function where to look

In this step, we select the table where the data is located, and tell excel to search in the leftmost column for the information we selected in the previous step.

For example, in this case we highlight the whole table from column A to column C. Excel will look for the information we told it to lookup in column A.

The screenshot shows an Excel spreadsheet with the following data and formula:

| Fruit     | In Stock? | Price (\$/lb) |
|-----------|-----------|---------------|
| Grapes    | Yes       | 7.25          |
| Mangos    | Yes       | 12.32         |
| Bananas   | Yes       | 5.42          |
| Pineapple | No        | 7.18          |
| Oranges   | Yes       | 8.99          |

Below the table, a smaller table is shown with the following data:

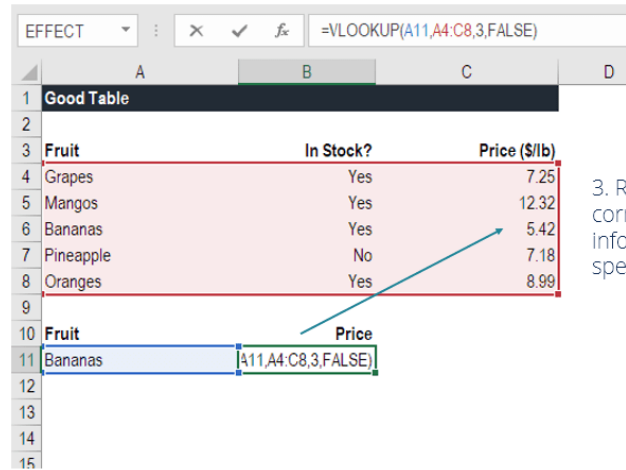
| Fruit   | Price                       |
|---------|-----------------------------|
| Bananas | =VLOOKUP(A11,A4:C8,3,FALSE) |

The formula bar at the top shows the formula: `=VLOOKUP(A11,A4:C8,3,FALSE)`. A blue arrow points from the text "2. In the left column of this table" to the leftmost column of the data table (column A).

#### Step 4: Tell Excel what column to output the data from

In this step, we need to tell Excel which column contains that data that we want to have as an output from the VLOOKUP. In order to do this, Excel needs a number that corresponds to the column number in the table.

In our example, the output data is located in the 3rd column of the table, so we enter the number "3" in the formula.



The screenshot shows an Excel spreadsheet with a formula bar at the top containing the formula `=VLOOKUP(A11,A4:C8,3,FALSE)`. The spreadsheet has columns A, B, C, and D, and rows 1 through 15. Row 1 is labeled "Good Table". Row 3 is the header for a table with columns "Fruit", "In Stock?", and "Price (\$/lb)". Rows 4 through 8 contain data for Grapes, Mangos, Bananas, Pineapple, and Oranges. Row 10 is a sub-header for "Fruit" and "Price". Row 11 contains "Bananas" in column A and the VLOOKUP formula in column B. A blue arrow points from the number "3" in the formula to the "Price" column header in row 3.

| Fruit     | In Stock? | Price (\$/lb) |
|-----------|-----------|---------------|
| Grapes    | Yes       | 7.25          |
| Mangos    | Yes       | 12.32         |
| Bananas   | Yes       | 5.42          |
| Pineapple | No        | 7.18          |
| Oranges   | Yes       | 8.99          |

3. Return the corresponding information in this specified column

### Step 5: Exact or approximate match

This final step is to tell Excel if you're looking for an exact or approximate match by entering "True" or "False" in the formula.

In our VLOOKUP example, we want an exact match, so we type "True" in the formula.

An approximate match would be useful when looking up an exact figure that might not be contained in the table, for example, the number 2.9585. In this case, Excel will look for the number closest to 2.9585, even though that number is not contained in the dataset. This will help prevent errors in the VLOOKUP formula.

Learn how to do this step by step in our [Free Excel Crash Course!](#)

### VLOOKUP in financial modeling and financial analysis

VLOOKUP formulas are often used in financial modeling and other types of financial analysis to make models more dynamic and incorporate multiple scenarios.

For example, imagine a financial model that included a debt schedule, and the company had three different scenarios for the interest rate: 3.0%, 4.0% and 5.0%. A VLOOKUP could look for a scenario, low, medium or high and output the corresponding interest rate into the financial model.

The screenshot shows an Excel spreadsheet with a formula bar at the top containing the formula `=VLOOKUP(H11,H6:J8,3,FALSE)`. The spreadsheet has columns G through M and rows 1 through 12. A table of interest rates is located in the range H6:J8:

| Debt Scenario | Rate |
|---------------|------|
| Low           | 3.0% |
| Medium        | 4.0% |
| High          | 5.0% |

Below this table, in the range H10:J12, there is another table for the financial model:

| Model Scenario | Interest                                   |
|----------------|--|
| Medium         | <code>{=VLOOKUP(H11,H6:J8,3,FALSE)}</code> |

As you can see in the example above, an Analyst can select the scenario they want and have the corresponding interest rate flow into the model from the VLOOKUP formula.

Learn how to do this step by step in our [Free Excel Crash Course!](#)

## Things to remember about the VLOOKUP Function

Here is an important list of things to remember about the Excel VLOOKUP Function:

1. When `range_lookup` is omitted, the VLOOKUP function will allow a non-exact match, but it will use an exact match if one exists.
2. The biggest limitation of the function is that it always looks right. It will get data from the columns to the right of the first column in the table.
3. If the lookup column contains duplicate values, VLOOKUP will match the first value only.
4. The function is not case-sensitive.
5. Suppose there's an existing VLOOKUP formula in a worksheet. In that scenario, formulas may break if we insert a column in the table. This is so as hard-coded column index values don't change automatically when columns are inserted or deleted.
6. VLOOKUP allows the use of wildcards, i.e., an asterisk (\*) and a question mark (?).
7. Suppose the table we are working with the function contains numbers entered as text. If we are simply retrieving numbers as text from a column in a table, it doesn't matter. But if the first column of the table contains numbers entered as text, we will get an #N/A! error if the lookup value is not also in text form.
8. #N/A! error – Occurs if the VLOOKUP function fails to find a match to the supplied `lookup_value`.
9. #REF! error – Occurs if either:
  - The `col_index_num` argument is greater than the number of columns in the supplied `table_array`; or
  - The formula attempted to reference cells that do not exist.
10. #VALUE! error – Occurs if either:
  - The `col_index_num` argument is less than 1 or is not recognized as a numeric value; or
  - The `range_lookup` argument is not recognized as one of the logical values TRUE or FALSE.

## Math

**ABS**

Find the absolute value of a number

number

### What is the ABSOLUTE Function in Excel (ABS)?

The ABSOLUTE function in Excel returns the absolute value of a number. The function converts negative numbers to positive numbers while positive numbers remain unaffected.

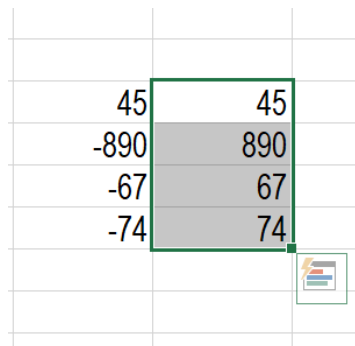
### Formula

**ABSOLUTE Value = ABS(number)**

Where **number** is the numeric value for which we need to calculate the Absolute value.

### How to use the ABSOLUTE Function in Excel?

Let's take a series of number to understand how this function can be used.



|      |     |
|------|-----|
| 45   | 45  |
| -890 | 890 |
| -67  | 67  |
| -74  | 74  |

In the screenshot above, we are given a series of numbers. When we use the ABSOLUTE function, we get following results:

1. For positive numbers, we get the same result. So 45 is returned as 45.
2. For negative numbers, the function returns absolute numbers. So for -890, -67, -74, we got 890, 67, 74.

To learn more, launch our [free Excel crash course](#) now!

## Examples of the ABSOLUTE Function in Excel

For our analysis, we want the difference between Series A and Series B as given below. Ideally, if you subtract Series A from Series B you might get negative numbers depending on the values. However, if you want absolute numbers in this scenario, we can use the ABSOLUTE function.

| D10              |          |          |                     |                    |
|------------------|----------|----------|---------------------|--------------------|
| fx =ABS(A10-B10) |          |          |                     |                    |
|                  | A        | B        | C                   | D                  |
| 9                | Series A | Series B | Series A - Series B | Using ABS Function |
| 10               | 10       | 8        | 2                   | 2                  |
| 11               | 6        | 8        | -2                  | 2                  |
| 12               | 7        | 9        | -2                  | 2                  |
| 13               | 5        | 8        | -3                  | 3                  |

The results returned using ABSOLUTE function would be absolute numbers. So ABS can be combined with other functions such as SUM, MAX, MIN, AVERAGE, etc. to calculate the absolute value for positive and negative numbers in Excel spreadsheets.

| E10                   |          |          |                     |                    |   |
|-----------------------|----------|----------|---------------------|--------------------|---|
| fx =ABS(MAX(B10,C10)) |          |          |                     |                    |   |
|                       | A        | B        | C                   | D                  | E |
| 9                     | Series A | Series B | Series A - Series B | Using ABS Function |   |
| 10                    | 10       | 8        | 2                   | 2                  | 8 |
| 11                    | 6        | 8        | -2                  | 2                  |   |
| 12                    | 7        | 9        | -2                  | 2                  |   |
| 13                    | 5        | 8        | -3                  | 3                  |   |

Let's see few examples of how ABS can be used with other Excel Functions.

## 1. SUMIF and ABS

We all are aware that SUMIF would sum up values if certain criteria within the range given are met. Let's assume we've been given few numbers in Column A and Column B as below:

|   | A  | B   |
|---|----|-----|
| 1 |    |     |
| 2 | 15 | 7   |
| 3 | 6  | 9   |
| 4 | -9 | -10 |
| 5 |    |     |

Now, I wish to subtract all negative numbers in Column B from all positive numbers of Column A. I want the result to be an absolute number. So I can use the ABS function along with SUMIF in the following manner:

|   | A  | B   | C | D | E | F |
|---|----|---|---|---|---|---|
| 1 |    |   |   |   |   |   |
| 2 | 15 | 7   |   |   |   |   |
| 3 | 6  | 9   |   |   |   |   |
| 4 | -9 | -10   |   |   |   |   |
| 5 |    |   |   |   |   |   |
| 6 |    | =ABS((SUMIF(A2:A4,">0")+(SUMIF(B2:B4,"<0")))) |   |   |   |   |
| 7 |    |   |   |   |   |   |

The result is 79. Excel added 15 and 6 from Column A and subtracted 100 from Column B to give us 79, as we used ABS function instead of -79.

## 2. SUM ARRAY Formula and ABS Function

The Excel array formulas help us to do multiple calculations for a given array or column of values. We can use SUM ARRAY along with ABS to get the absolute value of a series of numbers in column or row. Suppose we are given a few numbers as below, so in this scenario, the SUM array formula for absolute values would be =SUM(ABS(A2:A6)).

Now, select cell A7 in your spreadsheet, and enter the formula '=SUM(ABS(A2:A6))'. After entering the formula in cell A7, press "Ctrl + Shift + Enter". Once we do this, the formula will have {} brackets around it as shown in the screenshot below.

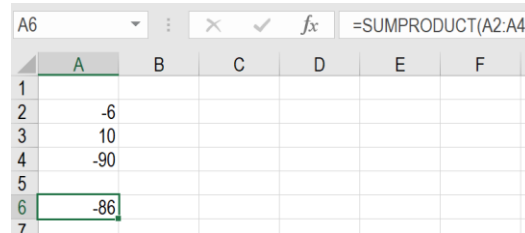
|   | A   | B | C | D | E | F |
|---|-----|---|---|---|---|---|
| 1 | 3   |   |   |   |   |   |
| 2 | 5   |   |   |   |   |   |
| 3 | 8   |   |   |   |   |   |
| 4 | 9   |   |   |   |   |   |
| 5 | -10 |   |   |   |   |   |
| 6 | -9  |   |   |   |   |   |
| 7 | 44  |   |   |   |   |   |

As seen in the screenshot above, the array formula also returned the value 44 in cell A7, which is the absolute value of the data entered in cells A2:A6.



### 3. SUMPRODUCT Formula and ABS Function

The SUMPRODUCT function allows us to include the ABS function to provide absolute numbers. Suppose we are given the following data. If we would just use the SUMPRODUCT formula, we would get a negative number as shown below:

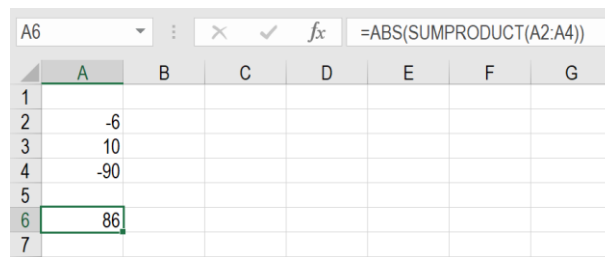


The screenshot shows an Excel spreadsheet with the following data in column A:

|   | A   | B | C | D | E | F |
|---|-----|---|---|---|---|---|
| 1 |     |   |   |   |   |   |
| 2 | -6  |   |   |   |   |   |
| 3 | 10  |   |   |   |   |   |
| 4 | -90 |   |   |   |   |   |
| 5 |     |   |   |   |   |   |
| 6 | -86 |   |   |   |   |   |
| 7 |     |   |   |   |   |   |

The formula bar shows the formula `=SUMPRODUCT(A2:A4)` and the result in cell A6 is -86.

However, using ABS Function, we can get the absolute number as result. The formula to be used would be:



The screenshot shows an Excel spreadsheet with the same data in column A as above:

|   | A   | B | C | D | E | F | G |
|---|-----|---|---|---|---|---|---|
| 1 |     |   |   |   |   |   |   |
| 2 | -6  |   |   |   |   |   |   |
| 3 | 10  |   |   |   |   |   |   |
| 4 | -90 |   |   |   |   |   |   |
| 5 |     |   |   |   |   |   |   |
| 6 | 86  |   |   |   |   |   |   |
| 7 |     |   |   |   |   |   |   |

The formula bar shows the formula `=ABS(SUMPRODUCT(A2:A4))` and the result in cell A6 is 86.

To learn more, launch our free [Excel crash course](#) now!

#### ABS as a VBA Function

If we wish to use the ABSOLUTE function in Excel VBA code, it can be used in the following manner. Let's assume I need ABS of -600 so the code would be:

#### Dim LNumber As Double

```
LNumber = ABS(-600)
```

Now in the above code, the variable known as LNumber would now contain the value of 600.

**SUMIF** Find the absolute value of a number `range` `criteria` `sum_range`

### What is the SUMIF Function?

The SUMIF function is categorized under **Math and Trigonometry functions**. It will sum up cells that meet the given criteria. The criteria are based on dates, numbers, and text. It supports logical operators such as (>, <, <>, =) and also wildcards (\*, ?). This guide to the SumIf Excel function will show you how to use it, step-by-step.

As a **financial analyst**, SUMIF is a frequently used function. Suppose we are given a table listing the consignments of vegetables from different suppliers. The names of the vegetable, names of suppliers and quantity are in column A, column B, and column C, respectively. In such scenario, we can use the SUMIF function to find out the sum of the amount related to a particular vegetable from a specific supplier.

### Formula

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

The formula uses the following arguments:

1. **Range** (required argument) – It is the range of cells that we want to apply the criteria against.
2. **Criteria** (required argument) – It is the criteria which are used to determine which cells need to be added.

When we provide the criteria argument, it can either be:

- A numeric value (which may be an integer, decimal, date, time, or logical value) (e.g. 10, 01/01/2018, TRUE) or
  - A text string (e.g. "Text", "Thursday") or
  - An expression (e.g. ">12", "<>0").
3. **Sum\_range** (optional argument) – It is an array of numeric values (or cells containing numeric values) that are to be added together if the corresponding range entry satisfies the supplied criteria. If the [sum\_range] argument is omitted, the values from the range argument are summed instead.

### How to use the SUMIF Excel Function

To understand the uses of the SUMIF function, let us consider a few examples:

## Example 1

Suppose we are given the following data:

|    | A | B              | C             | D            |
|----|---|----------------|---------------|--------------|
| 1  |   |                |               |              |
| 2  |   | SUMIF Function |               |              |
| 3  |   |                |               |              |
| 4  |   | <b>Month</b>   | <b>Region</b> | <b>Sales</b> |
| 5  |   | Jan            | East          | 450          |
| 6  |   | Jan            | South         | 650          |
| 7  |   | Feb            | West          | 900          |
| 8  |   | Jan            | East          | 450          |
| 9  |   | Feb            | South         | 900          |
| 10 |   | March          | North         | 890          |
| 11 |   | March          | South         | 765          |
| 12 |   | March          | East          | 435          |

We wish to find total sales for the East region and the total sales for February. The formula to use to get the total sales for East is:

The screenshot shows the Excel formula bar with the formula `=SUMIF(C5:C12,"East",D5:D12)`. Below the formula bar, the table from the previous image is shown with the 'Region' and 'Sales' columns highlighted in red. The 'Sales' values for the 'East' region are 450, 450, and 435. The formula bar also shows the result of the formula: `Total Sales for East =SUMIF(C5:C12,"East",D5:D12)`.

|    | A | B                           | C                            | D            | E | F |
|----|---|-----------------------------|------------------------------|--------------|---|---|
| 1  |   |                             |                              |              |   |   |
| 2  |   | SUMIF Function              |                              |              |   |   |
| 3  |   |                             |                              |              |   |   |
| 4  |   | <b>Month</b>                | <b>Region</b>                | <b>Sales</b> |   |   |
| 5  |   | Jan                         | East                         | 450          |   |   |
| 6  |   | Jan                         | South                        | 650          |   |   |
| 7  |   | Feb                         | West                         | 900          |   |   |
| 8  |   | Jan                         | East                         | 450          |   |   |
| 9  |   | Feb                         | South                        | 900          |   |   |
| 10 |   | March                       | North                        | 890          |   |   |
| 11 |   | March                       | South                        | 765          |   |   |
| 12 |   | March                       | East                         | 435          |   |   |
| 13 |   |                             |                              |              |   |   |
| 14 |   |                             |                              |              |   |   |
| 15 |   | <b>Total Sales for East</b> | =SUMIF(C5:C12,"East",D5:D12) |              |   |   |

Text criteria, or criteria that includes math symbols, must be enclosed in double quotation marks (" ").

We get the result below:

|    | A | B                           | C                     | D            | E | F |
|----|---|-----------------------------|-----------------------|--------------|---|---|
| 1  |   |                             |                       |              |   |   |
| 2  |   |                             | <b>SUMIF Function</b> |              |   |   |
| 3  |   |                             |                       |              |   |   |
| 4  |   | <b>Month</b>                | <b>Region</b>         | <b>Sales</b> |   |   |
| 5  |   | Jan                         | East                  | 450          |   |   |
| 6  |   | Jan                         | South                 | 650          |   |   |
| 7  |   | Feb                         | West                  | 900          |   |   |
| 8  |   | Jan                         | East                  | 450          |   |   |
| 9  |   | Feb                         | South                 | 900          |   |   |
| 10 |   | March                       | North                 | 890          |   |   |
| 11 |   | March                       | South                 | 765          |   |   |
| 12 |   | March                       | East                  | 435          |   |   |
| 13 |   |                             |                       |              |   |   |
| 14 |   |                             |                       |              |   |   |
| 15 |   | <b>Total Sales for East</b> | <b>1,335</b>          |              |   |   |

The formula for total sales in February is:

|    | A | B                           | C                                    | D            | E | F |
|----|---|-----------------------------|--------------------------------------|--------------|---|---|
| 1  |   |                             |                                      |              |   |   |
| 2  |   |                             | <b>SUMIF Function</b>                |              |   |   |
| 3  |   |                             |                                      |              |   |   |
| 4  |   | <b>Month</b>                | <b>Region</b>                        | <b>Sales</b> |   |   |
| 5  |   | Jan                         | East                                 | 450          |   |   |
| 6  |   | Jan                         | South                                | 650          |   |   |
| 7  |   | Feb                         | West                                 | 900          |   |   |
| 8  |   | Jan                         | East                                 | 450          |   |   |
| 9  |   | Feb                         | South                                | 900          |   |   |
| 10 |   | March                       | North                                | 890          |   |   |
| 11 |   | March                       | South                                | 765          |   |   |
| 12 |   | March                       | East                                 | 435          |   |   |
| 13 |   |                             |                                      |              |   |   |
| 14 |   |                             |                                      |              |   |   |
| 15 |   | <b>Total Sales for East</b> | <b>1,335</b>                         |              |   |   |
| 16 |   | <b>Total Sales for Feb</b>  | <b>=SUMIF(B5:B12,\"Feb\",D5:D12)</b> |              |   |   |

We get the result below:

|    | A | B                           | C                     | D            | E | F | G |
|----|---|-----------------------------|-----------------------|--------------|---|---|---|
| 1  |   |                             |                       |              |   |   |   |
| 2  |   |                             | <b>SUMIF Function</b> |              |   |   |   |
| 3  |   |                             |                       |              |   |   |   |
| 4  |   | <b>Month</b>                | <b>Region</b>         | <b>Sales</b> |   |   |   |
| 5  |   | Jan                         | East                  | 450          |   |   |   |
| 6  |   | Jan                         | South                 | 650          |   |   |   |
| 7  |   | Feb                         | West                  | 900          |   |   |   |
| 8  |   | Jan                         | East                  | 450          |   |   |   |
| 9  |   | Feb                         | South                 | 900          |   |   |   |
| 10 |   | March                       | North                 | 890          |   |   |   |
| 11 |   | March                       | South                 | 765          |   |   |   |
| 12 |   | March                       | East                  | 435          |   |   |   |
| 13 |   |                             |                       |              |   |   |   |
| 14 |   |                             |                       |              |   |   |   |
| 15 |   | <b>Total Sales for East</b> |                       | <b>1,335</b> |   |   |   |
| 16 |   | <b>Total Sales for Feb</b>  |                       | <b>1,800</b> |   |   |   |

### Few notes about the SUMIF Excel Function

1. #VALUE! error – Occurs when the criteria provided is a text string that is more than 255 characters long.
2. When sum\_range is omitted, the cells in range will be summed.
3. The following wildcards can be used in text-related criteria:
  - ? – matches any single character
  - \* – matches any sequence of characters
4. To find a literal question mark or asterisk, use a tilde (~) in front of the question mark or asterisk (i.e. ~?, ~\*).

[Click here to download the sample Excel file](#)

## Statistical

**AVERAGE** Get the average of a group of numbers

### How to calculate AVERAGE in Excel?

The AVERAGE function is categorized under Statistical functions. It will return the average value of given series of numbers in Excel. It is used to calculate the arithmetic mean of a given set of arguments in Excel. This guide will show you step-by-step how to calculate the average in Excel.

As a financial analyst, the function is useful in finding out the average (mean) of a series of numbers. For example, we can find out the average sales for the last 12 months for a business.

### Formula

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

The function uses the following arguments:

1. **Number1** (required argument) – It is the first number or a cell reference or a range for which we want the average.
2. **Number2** (optional argument) – They are the additional numbers, cell references or a range for which we want the average. A maximum of 255 numbers is allowed.

### How to use AVERAGE Function in Excel?

To understand the uses of the AVERAGE function, let us consider a few examples:

## Example 1 – Average in Excel

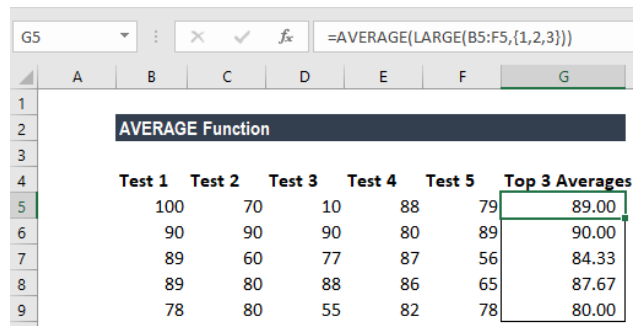
Suppose we are given the following data:

|   | A | B                       | C             | D             | E             | F             | G                     | H |
|---|---|-------------------------|---------------|---------------|---------------|---------------|-----------------------|---|
| 1 |   |                         |               |               |               |               |                       |   |
| 2 |   | <b>AVERAGE Function</b> |               |               |               |               |                       |   |
| 3 |   |                         |               |               |               |               |                       |   |
| 4 |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> | <b>Test 4</b> | <b>Test 5</b> | <b>Top 3 Averages</b> |   |
| 5 |   | 100                     | 70            | 10            | 88            | 79            | 89.00                 |   |
| 6 |   | 90                      | 90            | 90            | 80            | 89            | 90.00                 |   |
| 7 |   | 89                      | 60            | 77            | 87            | 56            | 84.33                 |   |
| 8 |   | 89                      | 80            | 88            | 86            | 65            | 87.67                 |   |
| 9 |   | 78                      | 80            | 55            | 82            | 78            | 80.00                 |   |

We wish to find out the top 3 scores in above data set. The formula to use will be:

|    | A | B                       | C             | D             | E             | F             | G                              | H | I |
|----|---|-------------------------|---------------|---------------|---------------|---------------|--------------------------------|---|---|
| 1  |   |                         |               |               |               |               |                                |   |   |
| 2  |   | <b>AVERAGE Function</b> |               |               |               |               |                                |   |   |
| 3  |   |                         |               |               |               |               |                                |   |   |
| 4  |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> | <b>Test 4</b> | <b>Test 5</b> | <b>Top 3 Averages</b>          |   |   |
| 5  |   | 100                     | 70            | 10            | 88            | 79            | =AVERAGE(LARGE(B5:F5,{1,2,3})) |   |   |
| 6  |   | 90                      | 90            | 90            | 80            | 89            | 90.00                          |   |   |
| 7  |   | 89                      | 60            | 77            | 87            | 56            | 84.33                          |   |   |
| 8  |   | 89                      | 80            | 88            | 86            | 65            | 87.67                          |   |   |
| 9  |   | 78                      | 80            | 55            | 82            | 78            | 80.00                          |   |   |
| 10 |   |                         |               |               |               |               |                                |   |   |

We get the result below:



|   | A | B                       | C             | D             | E             | F             | G                     |
|---|---|-------------------------|---------------|---------------|---------------|---------------|-----------------------|
| 1 |   |                         |               |               |               |               |                       |
| 2 |   | <b>AVERAGE Function</b> |               |               |               |               |                       |
| 3 |   |                         |               |               |               |               |                       |
| 4 |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> | <b>Test 4</b> | <b>Test 5</b> | <b>Top 3 Averages</b> |
| 5 |   | 100                     | 70            | 10            | 88            | 79            | 89.00                 |
| 6 |   | 90                      | 90            | 90            | 80            | 89            | 90.00                 |
| 7 |   | 89                      | 60            | 77            | 87            | 56            | 84.33                 |
| 8 |   | 89                      | 80            | 88            | 86            | 65            | 87.67                 |
| 9 |   | 78                      | 80            | 55            | 82            | 78            | 80.00                 |

In above formula, the LARGE function retrieved the top nth values from a set of values. So, we got the top 3 values as we used the array constant {1,2,3} into LARGE for the second argument.

Later, the AVERAGE function returned the average of the values. As the function can automatically handle array results, so we don't need not use Ctrl+Shift+Enter to enter the formula.



## Example 2 – Average in Excel

Suppose we are given the data below:

|   | A | B                       | C             | D             |
|---|---|-------------------------|---------------|---------------|
| 1 |   |                         |               |               |
| 2 |   | <b>AVERAGE Function</b> |               |               |
| 3 |   |                         |               |               |
| 4 |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> |
| 5 |   | 125                     | 150           | 200           |
| 6 |   | 90                      |               | 90            |
| 7 |   | 85                      | 85            | #N/A          |
| 8 |   | 0                       | 85            | 75            |

The formula to use is shown below:

The screenshot shows the Excel formula bar with the formula `=AVERAGE(B5:D5)` entered. Below the formula bar, the spreadsheet shows the same data as in the previous table, but with an additional column labeled 'Average' in column E. The values in column E are 90.00, #N/A, and 53.33, corresponding to rows 5, 6, and 7 respectively. The formula bar also shows the formula `=AVERAGE(B5:D5)` being applied to the range B5:D5.

|   | A | B                       | C             | D             | E               | F | G |
|---|---|-------------------------|---------------|---------------|-----------------|---|---|
| 1 |   |                         |               |               |                 |   |   |
| 2 |   | <b>AVERAGE Function</b> |               |               |                 |   |   |
| 3 |   |                         |               |               |                 |   |   |
| 4 |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> | <b>Average</b>  |   |   |
| 5 |   | 125                     | 150           | 200           | =AVERAGE(B5:D5) |   |   |
| 6 |   | 90                      |               | 90            | 90.00           |   |   |
| 7 |   | 85                      | 85            | #N/A          | #N/A            |   |   |
| 8 |   | 0                       | 85            | 75            | 53.33           |   |   |

We get the following result:

|   | A | B                       | C             | D             | E              | F | G |
|---|---|-------------------------|---------------|---------------|----------------|---|---|
| 1 |   |                         |               |               |                |   |   |
| 2 |   | <b>AVERAGE Function</b> |               |               |                |   |   |
| 3 |   |                         |               |               |                |   |   |
| 4 |   | <b>Test 1</b>           | <b>Test 2</b> | <b>Test 3</b> | <b>Average</b> |   |   |
| 5 |   | 125                     | 150           | 200           | 158.33         |   |   |
| 6 |   | 90                      |               | 90            | 90.00          |   |   |
| 7 |   | 85                      | 85            | #N/A          | #N/A           |   |   |
| 8 |   | 0                       | 85            | 75            | 53.33          |   |   |

### A few notes about the AVERAGE Function

1. The AVERAGE function ignores empty cells.
2. If a range or cell reference argument contains text, logical values, or empty cells, those values are ignored. However, cells with the value zero are included.
3. Arguments that are error values or text that cannot be translated into numbers cause errors in the function.

[Click here to download the sample Excel file](#)

**CORREL** Calculate the correlation coefficient between two variables

array1

array2

### What is the CORREL Function?

The CORREL function is categorized under **Statistical functions**. It will calculate the correlation coefficient between two variables.

As a **financial analyst**, the CORREL function is very useful when we want to find the correlation between two variables, i.e., the correlation between a particular stock and the market index.

### Correlation Formula

**=CORREL(array1, array2)**

The CORREL function uses the following arguments:

1. **Array1** (required argument) – It is the set of independent variables. It is a cell range of values.
2. **Array2** (required argument) – It is the set of dependent variables. It is the second cell range of values.

The equation for the correlation coefficient is:

$$\text{Correl}(X, Y) = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{\sum (x - \bar{x})^2 \sum (y - \bar{y})^2}}$$

Where:

$\bar{x}$  and  $\bar{y}$  are the sample means AVERAGE(array1) and AVERAGE(array2).

So, if the value of r is close to +1, it indicates a strong positive correlation, and if r is close to -1, it shows a strong negative correlation.

### How to use CORREL Function in Excel?

The CORREL function was introduced in Excel 2007 and is available in all subsequent Excel versions. To understand the uses of the function, let us consider an example:

## Correlation Example

Suppose we are given data about the weekly returns of stock A and percentage of change in the market index (S&P 500):

|    | A | B                      | C                       | D                                | E |
|----|---|------------------------|-------------------------|----------------------------------|---|
| 1  |   |                        |                         |                                  |   |
| 2  |   | <b>CORREL Function</b> |                         |                                  |   |
| 3  |   |                        |                         |                                  |   |
| 4  |   | <b>Week</b>            | <b>Stock A % Change</b> | <b>S&amp;P 500 weekly change</b> |   |
| 5  |   | 1                      | 3.50%                   | 1.25%                            |   |
| 6  |   | 2                      | 2.57%                   | 1.55%                            |   |
| 7  |   | 3                      | -0.50%                  | -1.14%                           |   |
| 8  |   | 4                      | 4.52%                   | 3.12%                            |   |
| 9  |   | 5                      | -1.27%                  | -0.85%                           |   |
| 10 |   | 6                      | 2.01%                   | 1.55%                            |   |
| 11 |   | 7                      | 2.63%                   | 0.95%                            |   |
| 12 |   | 8                      | -1.16%                  | -1.34%                           |   |
| 13 |   | 9                      | 0.41%                   | -0.30%                           |   |
| 14 |   | 10                     | 5.37%                   | 1.65%                            |   |
| 15 |   | 11                     | -1.10%                  | -1.34%                           |   |
| 16 |   | 12                     | 0.41%                   | 0.71%                            |   |
| 17 |   | 13                     | 0.83%                   | -0.14%                           |   |
| 18 |   | 14                     | 0.90%                   | 0.71%                            |   |
| 19 |   | 15                     | 2.01%                   | 1.50%                            |   |
| 20 |   | 16                     | 2.10%                   | 1.60%                            |   |
| 21 |   | 17                     | 3.00%                   | 1.22%                            |   |
| 22 |   | 18                     | -2.27%                  | -3.50%                           |   |
| 23 |   | 19                     | 1.06%                   | 0.50%                            |   |
| 24 |   | 20                     | -0.08%                  | -1%                              |   |
| 25 |   |                        |                         |                                  |   |

The formula used to find the correlation is:

SUM    X    ✓    fx    =CORREL(C5:C24,D5:D24)

| Week | Stock A % Change | S&P 500 weekly change |
|------|------------------|-----------------------|
| 1    | 3.50%            | 1.25%                 |
| 2    | 2.57%            | 1.55%                 |
| 3    | -0.50%           | -1.14%                |
| 4    | 4.52%            | 3.12%                 |
| 5    | -1.27%           | -0.85%                |
| 6    | 2.01%            | 1.55%                 |
| 7    | 2.63%            | 0.95%                 |
| 8    | -1.16%           | -1.34%                |
| 9    | 0.41%            | -0.30%                |
| 10   | 5.37%            | 1.65%                 |
| 11   | -1.10%           | -1.34%                |
| 12   | 0.41%            | 0.71%                 |
| 13   | 0.83%            | -0.14%                |
| 14   | 0.90%            | 0.71%                 |
| 15   | 2.01%            | 1.50%                 |
| 16   | 2.10%            | 1.60%                 |
| 17   | 3.00%            | 1.22%                 |
| 18   | -2.27%           | -3.50%                |
| 19   | 1.06%            | 0.50%                 |
| 20   | -0.08%           | -1%                   |

CORREL    =CORREL(C5:C24,D5:D24)

We get the result below:

The screenshot shows an Excel spreadsheet with the following data:

| Week | Stock A % Change | S&P 500 weekly change |
|------|------------------|-----------------------|
| 1    | 3.50%            | 1.25%                 |
| 2    | 2.57%            | 1.55%                 |
| 3    | -0.50%           | -1.14%                |
| 4    | 4.52%            | 3.12%                 |
| 5    | -1.27%           | -0.85%                |
| 6    | 2.01%            | 1.55%                 |
| 7    | 2.63%            | 0.95%                 |
| 8    | -1.16%           | -1.34%                |
| 9    | 0.41%            | -0.30%                |
| 10   | 5.37%            | 1.65%                 |
| 11   | -1.10%           | -1.34%                |
| 12   | 0.41%            | 0.71%                 |
| 13   | 0.83%            | -0.14%                |
| 14   | 0.90%            | 0.71%                 |
| 15   | 2.01%            | 1.50%                 |
| 16   | 2.10%            | 1.60%                 |
| 17   | 3.00%            | 1.22%                 |
| 18   | -2.27%           | -3.50%                |
| 19   | 1.06%            | 0.50%                 |
| 20   | -0.08%           | -1%                   |

The CORREL function result is 0.896907499.

The result indicates a strong positive correlation.

### Things to remember about the CORREL Function

1. #N/A error – Occurs if the given arrays are of different lengths. So, if Array1 and Array2 contain different numbers of data points, CORREL will return the #N/A error value.
2. #DIV/0 error – Occurs if either of the given arrays are empty or if the standard deviation of their values equals zero.
3. If an array or reference argument contains text, logical values, or empty cells, the values are ignored; however, cells with the value zero are included.
4. The CORREL function is exactly same as the PEARSON Function, except that, in earlier versions of Excel (earlier than Excel 2003), the PEARSON function may exhibit some rounding errors. Hence, it is advisable to use the CORREL function in earlier versions of Excel. In later versions of Excel, both functions should give the same results.

[Click here to download the sample Excel file](#)

**COUNT** Count numbers

value1

value2

...

### What is the COUNT Function?

The COUNT Function is a **Statistical function**. The function helps count the number of cells that contain a number, as well as the number of arguments that contain numbers. It will also count numbers in any given array. It was introduced in Excel in 2000.

As a **financial analyst**, the COUNT function is useful in analyzing data if we wish to keep a count of cells in given range.

### Formula

**=COUNT(value1, value2....)**

Where:

**Value1** (required argument) – The first item or cell reference or range for which we wish to count numbers.

**Value2...** (optional argument) – We can add up to 255 additional items, cell references, or ranges within which we wish to count numbers.

Remember this function will count only numbers and ignore everything else.

### How to use the COUNT Function in Excel?

To understand the uses of the COUNT function, let us consider few examples:



### Example 1

Let us see the results that we get using the data below:

| Data   | Formula           | Result |
|--------|-------------------|--------|
| 1/1/17 | =COUNT(1-25-2017) | 1      |
| 25     | =COUNT(25)        | 1      |
| TRUE   | =COUNT(TRUE)      | 0      |
| DIV/0! | =COUNT(#DIV/0!)   | 0      |
| 22.25  | =COUNT(22.25)     | 1      |
| 5.25   | =COUNT(5.25)      | 1      |

As seen above, the COUNT function ignored text or formula errors and counted numbers only.

The results we got in Excel are shown below:

| DATA      | FORMULA USED      | RESULT |
|-----------|-------------------|--------|
| 1/25/2017 | =COUNT(1-25-2017) | 1.00   |
| 25        | =COUNT(25)        | 1.00   |
| TRUE      | =COUNT(TRUE)      | -      |
| #DIV/0!   | =COUNT(#DIV/0!)   | -      |
| 22.25     | =COUNT(22.25)     | 1.00   |
| 5.25      | =COUNT(5.25)      | 1.00   |

## Few observations

1. Logical values and Errors are not counted by this function
2. As Excel stores dates as a serial number, the function returned 1 count for date.

The COUNT function can be used for an array. If we use the formula =COUNT(B5:B10), we will get the result 4 as shown below:

The screenshot shows an Excel spreadsheet with the following data and formulas:

| DATA      | FORMULA USED      | RESULT |
|-----------|-------------------|--------|
| 1/25/2017 | =COUNT(1-25-2017) | 1.00   |
| 25        | =COUNT(25)        | 1.00   |
| TRUE      | =COUNT(TRUE)      | -      |
| #DIV/0!   | =COUNT(#DIV/0!)   | -      |
| 22.25     | =COUNT(22.25)     | 1.00   |
| 5.25      | =COUNT(5.25)      | 1.00   |

Below the table, a formula bar shows the formula =COUNT(\$B\$5:\$B\$10) and the result 4.

## Example 2

Let's assume we imported data and wish to see the number of cells with numbers on it. The data given are shown below:

|    | A | B              | C         | D |
|----|---|----------------|-----------|---|
| 2  |   | COUNT Function |           |   |
| 3  |   |                |           |   |
| 4  |   | Data           |           |   |
| 5  |   | Apples         |           |   |
| 6  |   | Mangoes        |           |   |
| 7  |   | Kiwi           |           |   |
| 8  |   |                |           |   |
| 9  |   |                |           |   |
| 10 |   |                | 345       |   |
| 11 |   |                |           |   |
| 12 |   |                |           |   |
| 13 |   |                | 567       |   |
| 14 |   |                |           |   |
| 15 |   |                |           |   |
| 16 |   |                | 2/28/2013 |   |

To count the cells with numeric data, we will use the formula **COUNT(B4:B16)**.

|    | A | B              | C         | D | E      | F | G              | H       |
|----|---|----------------|-----------|---|--------|---|----------------|---------|
| 1  |   |                |           |   |        |   |                |         |
| 2  |   | COUNT Function |           |   |        |   |                |         |
| 3  |   |                |           |   |        |   |                |         |
| 4  |   | Data           |           |   |        |   |                |         |
| 5  |   | Apples         |           |   | DATA   |   | Count          | Remarks |
| 6  |   | Mangoes        |           |   | B4:B16 |   | =COUNT(B4:B16) | numeric |
| 7  |   | Kiwi           |           |   |        |   |                |         |
| 8  |   |                |           |   |        |   |                |         |
| 9  |   |                |           |   |        |   |                |         |
| 10 |   |                | 345       |   |        |   |                |         |
| 11 |   |                |           |   |        |   |                |         |
| 12 |   |                |           |   |        |   |                |         |
| 13 |   |                | 567       |   |        |   |                |         |
| 14 |   |                |           |   |        |   |                |         |
| 15 |   |                |           |   |        |   |                |         |
| 16 |   |                | 2/28/2013 |   |        |   |                |         |
| 17 |   |                |           |   |        |   |                |         |

We get 3 as the result, as shown below:

The screenshot shows an Excel spreadsheet with the following data:

|    | A | B                     | C         | D | E      | F | G     | H       |
|----|---|-----------------------|-----------|---|--------|---|-------|---------|
| 1  |   |                       |           |   |        |   |       |         |
| 2  |   | <b>COUNT Function</b> |           |   |        |   |       |         |
| 3  |   |                       |           |   |        |   |       |         |
| 4  |   | Data                  |           |   |        |   |       |         |
| 5  |   | Apples                |           |   | DATA   |   | Count | Remarks |
| 6  |   | Mangoes               |           |   | B4:B16 |   | 3     | numeric |
| 7  |   | Kiwi                  |           |   |        |   |       |         |
| 8  |   |                       |           |   |        |   |       |         |
| 9  |   |                       |           |   |        |   |       |         |
| 10 |   |                       | 345       |   |        |   |       |         |
| 11 |   |                       |           |   |        |   |       |         |
| 12 |   |                       |           |   |        |   |       |         |
| 13 |   |                       | 567       |   |        |   |       |         |
| 14 |   |                       |           |   |        |   |       |         |
| 15 |   |                       |           |   |        |   |       |         |
| 16 |   |                       | 2/28/2013 |   |        |   |       |         |
| 17 |   |                       |           |   |        |   |       |         |

The COUNT function is fully programmed. It counts the number of cells in a range that contain numbers and returns the result as shown above.

Suppose if we use formula COUNT(B5:B17,345). We will get the result below:

The screenshot shows an Excel spreadsheet with the following data:

|    | A | B                     | C         | D | E                  | F | G     | H       |
|----|---|-----------------------|-----------|---|--------------------|---|-------|---------|
| 2  |   | <b>COUNT Function</b> |           |   |                    |   |       |         |
| 3  |   |                       |           |   |                    |   |       |         |
| 4  |   | Data                  |           |   |                    |   |       |         |
| 5  |   | Apples                |           |   | DATA               |   | Count | Remarks |
| 6  |   | Mangoes               |           |   | B4:B16             |   | 3     | numeric |
| 7  |   | Kiwi                  |           |   | =COUNT(B5:B17,345) |   | 4     |         |
| 8  |   |                       |           |   |                    |   |       |         |
| 9  |   |                       |           |   |                    |   |       |         |
| 10 |   |                       | 345       |   |                    |   |       |         |
| 11 |   |                       |           |   |                    |   |       |         |
| 12 |   |                       |           |   |                    |   |       |         |
| 13 |   |                       | 567       |   |                    |   |       |         |
| 14 |   |                       |           |   |                    |   |       |         |
| 15 |   |                       |           |   |                    |   |       |         |
| 16 |   |                       | 2/28/2013 |   |                    |   |       |         |
| 17 |   |                       |           |   |                    |   |       |         |

You may be wondering that B10 contains 345 in the given range. So why the COUNT FUNCTION returned 4?

The reason is that in the COUNT function, all values in the formula are put side by side and then all numbers get counted. Therefore, the number "345" has nothing to do with the range. As a result, the formula will add the numbers of the two values in the formula.

### Example 3 – Using COUNT function with AVERAGE function

Suppose the prices of a certain commodity are given below:

|    | A | B                     | C        | D | E |
|----|---|-----------------------|----------|---|---|
| 1  |   |                       |          |   |   |
| 2  |   | <b>COUNT Function</b> |          |   |   |
| 3  |   |                       |          |   |   |
| 4  |   | 01/01/17              | \$ 22.50 |   |   |
| 5  |   | 01/02/17              | \$ 22.75 |   |   |
| 6  |   | 01/03/17              | \$ 22.00 |   |   |
| 7  |   | 01/04/17              | \$ 22.80 |   |   |
| 8  |   | 01/05/17              | \$ 22.65 |   |   |
| 9  |   | 01/06/17              | \$ 23.00 |   |   |
| 10 |   | 01/07/17              | \$ 23.10 |   |   |
| 11 |   | 01/08/17              | \$ 23.50 |   |   |
| 12 |   | 01/09/17              | \$ 23.55 |   |   |
| 13 |   | 01/10/17              | \$ 23.50 |   |   |
| 14 |   | 01/11/17              | \$ 23.45 |   |   |
| 15 |   | 01/12/17              | \$ 23.40 |   |   |

If we wish to find out the average price from January 8 to 12, we can use AVERAGE function along with COUNT and OFFSET functions.

The formula to use will be:

|    | A | B                     | C                                    | D | E | F | G | H | I |
|----|---|-----------------------|--------------------------------------|---|---|---|---|---|---|
| 2  |   | <b>COUNT Function</b> |                                      |   |   |   |   |   |   |
| 3  |   |                       |                                      |   |   |   |   |   |   |
| 4  |   | 01/01/17              | \$ 22.50                             |   |   |   |   |   |   |
| 5  |   | 01/02/17              | \$ 22.75                             |   |   |   |   |   |   |
| 6  |   | 01/03/17              | \$ 22.00                             |   |   |   |   |   |   |
| 7  |   | 01/04/17              | \$ 22.80                             |   |   |   |   |   |   |
| 8  |   | 01/05/17              | \$ 22.65                             |   |   |   |   |   |   |
| 9  |   | 01/06/17              | \$ 23.00                             |   |   |   |   |   |   |
| 10 |   | 01/07/17              | \$ 23.10                             |   |   |   |   |   |   |
| 11 |   | 01/08/17              | \$ 23.50                             |   |   |   |   |   |   |
| 12 |   | 01/09/17              | \$ 23.55                             |   |   |   |   |   |   |
| 13 |   | 01/10/17              | \$ 23.50                             |   |   |   |   |   |   |
| 14 |   | 01/11/17              | \$ 23.45                             |   |   |   |   |   |   |
| 15 |   | 01/12/17              | \$ 23.40                             |   |   |   |   |   |   |
| 16 |   |                       |                                      |   |   |   |   |   |   |
| 17 |   |                       | =AVERAGE(OFFSET(C4,COUNT(C:C),0,-5)) |   |   |   |   |   |   |
| 18 |   |                       |                                      |   |   |   |   |   |   |

The OFFSET function helped in creating dynamic rectangular ranges. By giving the starting reference B2, we specified the rows and columns the final range would include.

OFFSET would now return a range originating from the last entry in column B. Now the COUNT function is used for all of column B to get the required row offset. COUNT counts only numeric values, so the headings if any are automatically ignored.

There are 12 numerical values in column B so offset would resolve to OFFSET(B2,12,0,-5). With these values, OFFSET starts at B2, offset 12 rows to B13, then uses -5 to extend the rectangular range up "backward" 5 rows to create the range B9:B12.

Finally, OFFSET returns the range B9:B12 to the AVERAGE function, which computes the average of values in that range.

|    | A | B        | C              | D | E | F | G | H |
|----|---|----------|----------------|---|---|---|---|---|
| 2  |   |          | COUNT Function |   |   |   |   |   |
| 3  |   |          |                |   |   |   |   |   |
| 4  |   | 01/01/17 | \$ 22.50       |   |   |   |   |   |
| 5  |   | 01/02/17 | \$ 22.75       |   |   |   |   |   |
| 6  |   | 01/03/17 | \$ 22.00       |   |   |   |   |   |
| 7  |   | 01/04/17 | \$ 22.80       |   |   |   |   |   |
| 8  |   | 01/05/17 | \$ 22.65       |   |   |   |   |   |
| 9  |   | 01/06/17 | \$ 23.00       |   |   |   |   |   |
| 10 |   | 01/07/17 | \$ 23.10       |   |   |   |   |   |
| 11 |   | 01/08/17 | \$ 23.50       |   |   |   |   |   |
| 12 |   | 01/09/17 | \$ 23.55       |   |   |   |   |   |
| 13 |   | 01/10/17 | \$ 23.50       |   |   |   |   |   |
| 14 |   | 01/11/17 | \$ 23.45       |   |   |   |   |   |
| 15 |   | 01/12/17 | \$ 23.40       |   |   |   |   |   |
| 16 |   |          |                |   |   |   |   |   |
| 17 |   |          | \$ 23.48       |   |   |   |   |   |
| 18 |   |          |                |   |   |   |   |   |

**Things to remember in the COUNT Function**

1. If we wish to count logical values, we should use the COUNTA function.
2. The function belongs to COUNT function family. There are five variants of COUNT functions: COUNT, COUNTA, COUNTBLANK, COUNTIF, and COUNTIFS.
3. We need to use COUNTIF function or COUNTIFS function if we want to count only numbers that meet specific criteria.
4. If we wish to count based on certain criteria, then we should use COUNTIF.
5. The COUNT function doesn't count logical values TRUE or FALSE.

[Click here to download the sample Excel file](#)

**COUNTA** Count the number of non-blank cells

value1

value2

...

### What is COUNTA (Excel Countif Not Blank)?

The COUNTA Function is categorized under **Statistical functions** and will calculate the number of cells that are not blank within a given set of values. The COUNTA function is also commonly referred to as the Excel COUNTIF Not Blank formula.

As a **financial analyst**, the COUNTA function is useful if we wish to keep a count of cells in a given range. Apart from crunching numbers, we often need to count cells with values. In such scenario, the function can be useful.

### Excel Countif Not Blank Formula

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

The Excel countif not blank formula uses the following arguments:

1. **Value1** (required argument) – It is the value at which we evaluate the function.
2. **Value2** (optional argument) – Additional arguments that represent the values that we wish to count.

### Few notes about the arguments

1. We can enter up to 255 value arguments if we are using MS Excel 2007 or later. Earlier versions can handle 30 arguments only.
2. Value arguments can be values, array of values or references to cell ranges.

### How to use the COUNTA Function in Excel?

To understand the uses of the COUNTA function, let us consider a few examples:



### Example 1 – Excel Countif not blank

Suppose we are given the data below:

| Data   | Formula           | Result |
|--------|-------------------|--------|
| 2/1/17 | =COUNT(2-01-2017) | 1.00   |
| 35     | =COUNT(35)        | 1.00   |
| TRUE   | =COUNT(TRUE)      | 1.00   |
| DIV/0! | =COUNT(#DIV/0!)   | 1.00   |
| 22.25  | =COUNT(22.25)     | 1.00   |
| 5.25   | =COUNT(5.25)      | 1.00   |

As seen above, the COUNTA function will count text or formula errors. So unlike the COUNT function, which considers only numbers, COUNTA considers numbers, dates, text values, logical values, and errors).

We get the results below:

| DATA     | FORMULA USED      | RESULT |
|----------|-------------------|--------|
| 2/1/2017 | =COUNT(2-01-2017) | 1.00   |
| 35       | =COUNT(35)        | 1.00   |
| TRUE     | =COUNT(TRUE)      | 1.00   |
| #DIV/0!  | =COUNT(#DIV/0!)   | 1.00   |
| 22.25    | =COUNT(22.25)     | 1.00   |
| 5.25     | =COUNT(5.25)      | 1.00   |

The COUNTA function can be used for an array. If we enter the formula =COUNTA(B5:B10), we will get the result 6, as shown below:

| DATA     | FORMULA USED      | RESULT |
|----------|-------------------|--------|
| 2/1/2017 | =COUNT(2-01-2017) | 1.00   |
| 35       | =COUNT(35)        | 1.00   |
| TRUE     | =COUNT(TRUE)      | 1.00   |
| #DIV/0!  | =COUNT(#DIV/0!)   | 1.00   |
| 22.25    | =COUNT(22.25)     | 1.00   |
| 5.25     | =COUNT(5.25)      | 1.00   |

Formula: =COUNTA(\$B\$5:\$B\$10)

Result: 6

## Example 2 - Excel Countif not blank

Suppose we wish to count the number of cells that contain data in a given set, as shown below:

|    | A | B               | C   | D |
|----|---|-----------------|-----|---|
| 2  |   | COUNTA Function |     |   |
| 3  |   |                 |     |   |
| 4  |   | Data            |     |   |
| 5  |   | He-man          |     |   |
| 6  |   | Spinach         |     |   |
| 7  |   | Tomato          |     |   |
| 8  |   |                 |     |   |
| 9  |   |                 |     |   |
| 10 |   |                 | 345 |   |
| 11 |   |                 |     |   |
| 12 |   |                 | 250 |   |
| 13 |   |                 | 567 |   |
| 14 |   |                 |     |   |
| 15 |   |                 |     |   |
| 16 |   | 2/28/2018       |     |   |
| 17 |   |                 |     |   |

To count the cells with data, we will use the formula =COUNTA(B4:B16).

|    | A      | B               | C               | D | E | F |
|----|--------|-----------------|-----------------|---|---|---|
| 2  |        | COUNTA Function |                 |   |   |   |
| 3  |        |                 |                 |   |   |   |
| 4  |        | Data            |                 |   |   |   |
| 5  |        | He-man          |                 |   |   |   |
| 6  |        | Spinach         |                 |   |   |   |
| 7  |        | Tomato          |                 |   |   |   |
| 8  |        |                 |                 |   |   |   |
| 9  |        |                 |                 |   |   |   |
| 10 |        |                 | 345             |   |   |   |
| 11 |        |                 |                 |   |   |   |
| 12 |        |                 | 250             |   |   |   |
| 13 |        |                 | 567             |   |   |   |
| 14 |        |                 |                 |   |   |   |
| 15 |        |                 |                 |   |   |   |
| 16 |        | 2/28/2018       |                 |   |   |   |
| 17 |        |                 |                 |   |   |   |
| 18 | DATA   |                 | Count           |   |   |   |
| 19 | B4:B16 |                 | =COUNTA(B4:B16) |   |   |   |

We get 8 as the result, as the COUNTA function will not count cells that are absolutely empty cells.

|    | A | B                      | C     | D | E | F |
|----|---|------------------------|-------|---|---|---|
| 2  |   | <b>COUNTA Function</b> |       |   |   |   |
| 3  |   |                        |       |   |   |   |
| 4  |   | Data                   |       |   |   |   |
| 5  |   | He-man                 |       |   |   |   |
| 6  |   | Spinach                |       |   |   |   |
| 7  |   | Tomato                 |       |   |   |   |
| 8  |   |                        |       |   |   |   |
| 9  |   |                        |       |   |   |   |
| 10 |   | 345                    |       |   |   |   |
| 11 |   |                        |       |   |   |   |
| 12 |   | 250                    |       |   |   |   |
| 13 |   | 567                    |       |   |   |   |
| 14 |   |                        |       |   |   |   |
| 15 |   |                        |       |   |   |   |
| 16 |   | 2/28/2018              |       |   |   |   |
| 17 |   |                        |       |   |   |   |
| 18 |   | DATA                   | Count |   |   |   |
| 19 |   | B4:B16                 | 8     |   |   |   |

### Example 3 – Excel Countif not blank

Suppose we wish to count cells that are not equal to a range of certain things. We can use a combination of the COUNTA, COUNTIF and SUMPRODUCT functions to get the desired results.

Suppose we are given the data below:

|   | A | B               | C |
|---|---|-----------------|---|
| 1 |   |                 |   |
| 2 |   | COUNTA Function |   |
| 3 |   |                 |   |
| 4 |   | Strawberry      |   |
| 5 |   | Banana          |   |
| 6 |   | Kiwi            |   |
| 7 |   | Pear            |   |
| 8 |   | Rose            |   |
| 9 |   | Marigold        |   |

To count cells not equal to Rose and Marigold, we used the following formula:

The screenshot shows the Excel formula bar with the formula `=COUNTA(B4:B9)-SUMPRODUCT(COUNTIF(B4:B9,D5:D6))`. Below the formula bar, the worksheet view shows the data from the previous example. Cell C2 contains the text "COUNTA Function". Cell C11 contains the formula `=COUNTA(B4:B9)-SUMPRODUCT(COUNTIF(B4:B9,D5:D6))`. A blue selection box highlights the range B4:B9 (Strawberry, Banana, Kiwi, Pear, Rose, Marigold), and a red selection box highlights the range D5:D6 (Rose, Marigold).

We can list down the things we wish to exclude from counting. One other way to arrive at the same result is to use the formula **=COUNTIFS(B4:B9,"<>Rose" B4:B9,"<>Marigold")**.

However, the above formula would be tedious to use if the list contains many items. We need to add an additional range/criteria pair for each item that we don't wish to count.

Hence, we can use a reference list and use the

**=COUNTA(B4:B9)-SUMPRODUCT(COUNTIF(B4:B9,D5:D6))** formula.

The formula uses the range D5:D6 to hold values that we don't want to count. The formula starts by counting all values in the range being counted with COUNTA. Next, it generates a count of all things we don't want to count with COUNTIF. Lastly, SUMPRODUCT will sum up all items in the array, which returns 2. The number is then subtracted from the original total to yield the final result.

We get the result below:

|    | A | B           | C                      | D        | E | F | G | H | I |
|----|---|-------------|------------------------|----------|---|---|---|---|---|
| 1  |   |             |                        |          |   |   |   |   |   |
| 2  |   |             | <b>COUNTA Function</b> |          |   |   |   |   |   |
| 3  |   |             |                        |          |   |   |   |   |   |
| 4  |   | Strawberry  |                        | Things   |   |   |   |   |   |
| 5  |   | Banana      |                        | Rose     |   |   |   |   |   |
| 6  |   | Kiwi        |                        | Marigold |   |   |   |   |   |
| 7  |   | Pear        |                        |          |   |   |   |   |   |
| 8  |   | Rose        |                        |          |   |   |   |   |   |
| 9  |   | Marigold    |                        |          |   |   |   |   |   |
| 10 |   |             |                        |          |   |   |   |   |   |
| 11 |   | Total count | 4.00                   |          |   |   |   |   |   |

**Things to remember about the COUNTA Function**

1. If we wish to count logical values, we should use the COUNTA function, but if we want to count only cells that contain a number, we should use COUNT function.
2. The function belongs to the COUNT function family. There are five variants of COUNT functions: COUNT, COUNTA, COUNTBLANK, COUNTIF, and COUNTIFS.
3. We need to use the COUNTIF function or COUNTIFS function if we want to count only numbers that meet the specific criteria.
4. If we wish to count based on certain criteria, we should use COUNTIF.

[Click here to download the sample Excel file](#)

**COUNTIF** Count cells that match multiple criteria

range1

criteria1

range2

criteria2

...

## What is the COUNTIF Function?

The COUNTIF function will count the number of cells that meet a specific criterion. The function is categorized under [Statistical functions](#).

In [financial analysis](#), the COUNTIF function is quite helpful. Take for example when we need to count the number of times a salesperson exceeded his target. We can do this by using COUNTIF.

## Formula

**=COUNTIF(Range, criteria)**

The COUNTIF function uses only one argument:

1. **Range** (required argument) – It defines one or several cells that we wish to count. The range of cells are those cells that will be tested against the given criteria and counted if the criteria are satisfied.
2. **Criteria** – It is a condition defined by us. It is tested against each of the cells in the supplied range.

The given criteria can be any of the following:

1. It is a numeric value. It can be an integer, decimal, date, time, or logical value.
2. A text string. It may include wildcards. Wildcards can be a ? (question mark) or an \* (asterisk). A ? matches any single character, whereas, \* matches any sequence of characters. If we wish to actually find the ? (question mark) or \* (asterisk) character, we need to type the ~ symbol before this character in our search.

## How to use the COUNTIF Function in Excel?

As a worksheet function, the COUNTIF Function can be entered as part of a formula in a cell of a worksheet.

To understand the uses of the function, let us consider a few examples:

### Example 1

Suppose we are given a list of employees awarded who exceeded the sales target and won the Best Salesperson award for the year. Using COUNTIF, we can find out which person won the maximum number of awards.

|    | A | B           | C                               |
|----|---|-------------|---------------------------------|
| 3  |   |             |                                 |
| 4  |   | <b>Year</b> | <b>Sales person of the year</b> |
| 5  |   | 2010        | Steffi Graf                     |
| 6  |   | 2011        | William Mathew                  |
| 7  |   | 2012        | Cesar Soo                       |
| 8  |   | 2013        | Mattan Lurie                    |
| 9  |   | 2014        | William Mathew                  |
| 10 |   | 2015        | Mathew Globe                    |
| 11 |   | 2016        | William Mathew                  |
| 12 |   | 2017        | Mathew Globe                    |

The formula to use would be =COUNT(C5:C12, C5) to get the maximum count, as shown below:

|    | A | B           | C                               | D                           | E | F | G |
|----|---|-------------|---------------------------------|-----------------------------|---|---|---|
| 3  |   |             |                                 |                             |   |   |   |
| 4  |   | <b>Year</b> | <b>Sales person of the year</b> |                             |   |   |   |
| 5  |   | 2010        | Steffi Graf                     | =COUNTIF(\$C\$5:\$C\$12,C5) |   |   |   |
| 6  |   | 2011        | William Mathew                  | 3                           |   |   |   |
| 7  |   | 2012        | Cesar Soo                       | 1                           |   |   |   |
| 8  |   | 2013        | Mattan Lurie                    | 1                           |   |   |   |
| 9  |   | 2014        | William Mathew                  | 3                           |   |   |   |
| 10 |   | 2015        | Mathew Globe                    | 2                           |   |   |   |
| 11 |   | 2016        | William Mathew                  | 3                           |   |   |   |
| 12 |   | 2017        | Mathew Globe                    | 2                           |   |   |   |
| 13 |   |             |                                 |                             |   |   |   |



We will now drag the formula for all salespersons and get the results below:

The screenshot shows an Excel spreadsheet with the following data:

|    | A | B           | C                               | D | E | F |
|----|---|-------------|---------------------------------|---|---|---|
| 3  |   |             |                                 |   |   |   |
| 4  |   | <b>Year</b> | <b>Sales person of the year</b> |   |   |   |
| 5  |   | 2010        | Steffi Graf                     | 1 |   |   |
| 6  |   | 2011        | William Mathew                  | 3 |   |   |
| 7  |   | 2012        | Cesar Soo                       | 1 |   |   |
| 8  |   | 2013        | Mattan Lurie                    | 1 |   |   |
| 9  |   | 2014        | William Mathew                  | 3 |   |   |
| 10 |   | 2015        | Mathew Globe                    | 2 |   |   |
| 11 |   | 2016        | William Mathew                  | 3 |   |   |
| 12 |   | 2017        | Mathew Globe                    | 2 |   |   |
| 13 |   |             |                                 |   |   |   |

The formula bar shows the formula: `=COUNTIF($C$5:$C$12,C5)`

As seen above, the name William Mathew appeared thrice. Remember, a criterion is case insensitive. Even if we type "william mathew" as the criterion in the above formula, it will produce the same result.

## Example 2

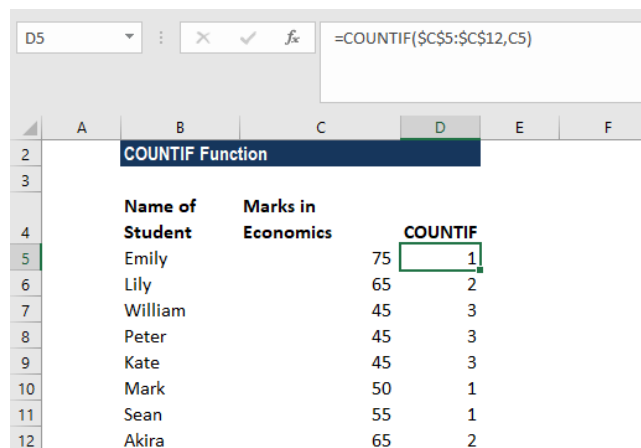
Let's see how this function behaves when we deal with numbers. The function works for numbers as efficiently as for text values. Suppose we are given the following data:

|    | A | B                      | C                         |
|----|---|------------------------|---------------------------|
| 2  |   | COUNTIF Function       |                           |
| 3  |   |                        |                           |
| 4  |   | <b>Name of Student</b> | <b>Marks in Economics</b> |
| 5  |   | Emily                  | 75                        |
| 6  |   | Lily                   | 65                        |
| 7  |   | William                | 45                        |
| 8  |   | Peter                  | 45                        |
| 9  |   | Kate                   | 45                        |
| 10 |   | Mark                   | 50                        |
| 11 |   | Sean                   | 55                        |
| 12 |   | Akira                  | 65                        |

Using the formula `=COUNTIF(C5:C12,C5)`, we can see that the function works well for numbers, too.

|    | A | B                      | C                         | D  | E | F |
|----|---|------------------------|---------------------------|--|---|---|
| 2  |   | COUNTIF Function       |                           |  |   |   |
| 3  |   |                        |                           |  |   |   |
| 4  |   | <b>Name of Student</b> | <b>Marks in Economics</b> | <b>COUNTIF</b>                           |   |   |
| 5  |   | Emily                  | 75                        | <code>=COUNTIF(\$C\$5:\$C\$12,C5)</code> |   |   |
| 6  |   | Lily                   | 65                        | 2  |   |   |
| 7  |   | William                | 45                        | 3  |   |   |
| 8  |   | Peter                  | 45                        | 3  |   |   |
| 9  |   | Kate                   | 45                        | 3  |   |   |
| 10 |   | Mark                   | 50                        | 1  |   |   |
| 11 |   | Sean                   | 55                        | 1  |   |   |
| 12 |   | Akira                  | 65                        | 2  |   |   |
| 13 |   |                        |                           |  |   |   |

The results we get are shown below:



|    | A | B                      | C                         | D              | E | F |
|----|---|------------------------|---------------------------|----------------|---|---|
| 2  |   | COUNTIF Function       |                           |                |   |   |
| 3  |   |                        |                           |                |   |   |
| 4  |   | <b>Name of Student</b> | <b>Marks in Economics</b> | <b>COUNTIF</b> |   |   |
| 5  |   | Emily                  | 75                        | 1              |   |   |
| 6  |   | Lily                   | 65                        | 2              |   |   |
| 7  |   | William                | 45                        | 3              |   |   |
| 8  |   | Peter                  | 45                        | 3              |   |   |
| 9  |   | Kate                   | 45                        | 3              |   |   |
| 10 |   | Mark                   | 50                        | 1              |   |   |
| 11 |   | Sean                   | 55                        | 1              |   |   |
| 12 |   | Akira                  | 65                        | 2              |   |   |

It shows that 3 students scored 45 in Economics.

### Example 3

Let's see how wildcards can be used with COUNTIF function. Suppose we employ several project managers who handle different projects. We wish to know how many projects are being handled by a particular manager. In such scenario, we can use the COUNTIF function.

We are given the following data:

|    | A | B                       | C                              | D |
|----|---|-------------------------|--------------------------------|---|
| 1  |   |                         |                                |   |
| 2  |   | <b>COUNTIF Function</b> |                                |   |
| 3  |   |                         |                                |   |
| 4  |   | <b>Project name</b>     | <b>Name of Project Manager</b> |   |
| 5  |   | Project A               | Mr. William                    |   |
| 6  |   | Project B               | Lily Emily                     |   |
| 7  |   | Project C               | William Shakespeare            |   |
| 8  |   | Project D               | Peter Parker                   |   |
| 9  |   | Project E               | William Shakespeare            |   |
| 10 |   | Project F               | Lily Emily                     |   |
| 11 |   | Project G               | Mr. William                    |   |
| 12 |   | Project H               | William Shakespeare            |   |

The name of Project Manager William Shakespeare is written in several ways. So we would enter "\*William\*" as the search criteria =COUNTIF(C5:C12, "\*William\*").

|    | A | B                       | C                              | D | E | F |
|----|---|-------------------------|--------------------------------|---|---|---|
| 1  |   |                         |                                |   |   |   |
| 2  |   | <b>COUNTIF Function</b> |                                |   |   |   |
| 3  |   |                         |                                |   |   |   |
| 4  |   | <b>Project name</b>     | <b>Name of Project Manager</b> |   |   |   |
| 5  |   | Project A               | Mr. William                    |   |   |   |
| 6  |   | Project B               | Lily Emily                     |   |   |   |
| 7  |   | Project C               | William Shakespeare            |   |   |   |
| 8  |   | Project D               | Peter Parker                   |   |   |   |
| 9  |   | Project E               | William Shakespeare            |   |   |   |
| 10 |   | Project F               | Lily Emily                     |   |   |   |
| 11 |   | Project G               | Mr. William                    |   |   |   |
| 12 |   | Project H               | William Shakespeare            |   |   |   |
| 13 |   |                         |                                |   |   |   |
| 14 |   | <b>COUNTIF</b>          | =COUNTIF(C5:C12, "*William*")  |   |   |   |
| 15 |   |                         |                                |   |   |   |

The results are as follows:

| Project name | Name of Project Manager |
|--------------|-------------------------|
| Project A    | Mr. William             |
| Project B    | Lily Emily              |
| Project C    | William Shakespeare     |
| Project D    | Peter Parker            |
| Project E    | William Shakespeare     |
| Project F    | Lily Emily              |
| Project G    | Mr. William             |
| Project H    | William Shakespeare     |

|                |   |
|----------------|---|
| <b>COUNTIF</b> | 5 |
|----------------|---|

If we need to match any single character, we need to enter a question mark instead of an asterisk.

#### Some notes about the COUNTIF Function:

1. The COUNTIF function is available from MS Excel 2000.
2. #VALUE! error – Occurs if the given criteria argument is a text string that is greater than 255 characters in length.

[Click here to download the sample Excel file](#)

**RANK** Rank a number against a range of numbers

number

array

order

## What is the RANK Function?

The RANK Function is categorized as a **Statistical function**. The function returns the statistical rank of a given value within a supplied array of value. Thus, it determines the position of a value in an array.

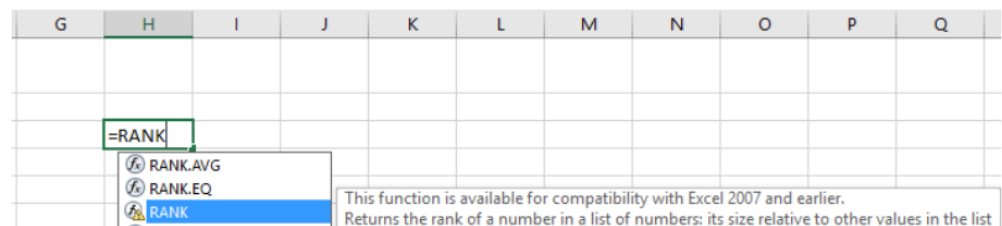
## Formula

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

The RANK function uses the following arguments:

1. **Number** (required argument) – It is the value for which we need to find the rank.
2. **Ref** (required argument) – It can be a list of or an array of or reference to numbers
3. **Order** (optional argument) – It is a number that specifies how the ranking will be done (ascending or descending order).
  - 0 – Is used for descending order
  - 1 – is used for ascending order
  - If we omit the argument, it will take a default value of 0 (descending order). It will take any non-zero value as the value 1 (ascending order).

Before we proceed, we need to know that the RANK function's been replaced by RANK.EQ and RANK.AVG. To enable backward compatibility, RANK still works in Excel 2016 (latest version), but it may not be available in future. If you type this function in Excel 2016, it will show a yellow triangle with an exclamation point.



## How to use the RANK Function in Excel?

As a worksheet function, RANK can be entered as part of a formula in a cell of a worksheet. To understand the uses of the function, let's consider a few examples:

### Example 1

Assuming we are given a list of employees with their respective medical reimbursement expenses. We wish to rank them according to total expenditure.

|    | A | B                       | C                                    |
|----|---|-------------------------|--------------------------------------|
| 1  |   |                         |                                      |
| 2  |   | <b>RANK Function</b>    |                                      |
| 3  |   |                         |                                      |
| 4  |   | <b>Name of Employee</b> | <b>Medical reimbursement expense</b> |
| 5  |   | John                    | 100                                  |
| 6  |   | Mathew                  | 200                                  |
| 7  |   | Tom                     | 25                                   |
| 8  |   | Mark                    | 360                                  |
| 9  |   | Lily                    | 100                                  |
| 10 |   | Marc                    | 560                                  |
| 11 |   |                         |                                      |

To rank in descending order, we will use the formula  
**=RANK(B2,(\$C\$5:\$C\$10),0)**, as shown below:

|    | A | B                       | C                                    | D                            | E               | F |
|----|---|-------------------------|--------------------------------------|------------------------------|-----------------|---|
| 1  |   |                         |                                      |                              |                 |   |
| 2  |   | <b>RANK Function</b>    |                                      |                              |                 |   |
| 3  |   |                         |                                      |                              |                 |   |
| 4  |   | <b>Name of Employee</b> | <b>Medical reimbursement expense</b> | <b>Ranks</b>                 | <b>New Rank</b> |   |
| 5  |   | John                    | 100                                  | =RANK(C5,(\$C\$5:\$C\$10),0) |                 |   |
| 6  |   | Mathew                  | 200                                  | 3                            | 3               |   |
| 7  |   | Tom                     | 25                                   | 6                            | 6               |   |
| 8  |   | Mark                    | 360                                  | 2                            | 2               |   |
| 9  |   | Lily                    | 100                                  | 4                            | 5               |   |
| 10 |   | Marc                    | 560                                  | 1                            | 1               |   |
| 11 |   |                         |                                      |                              |                 |   |

The result we get is shown below:

The screenshot shows the Excel formula bar with the formula `=RANK(C5,($C$5:$C$10),0)`. Below it is a table titled "RANK Function" with the following data:

| Name of Employee | Medical reimbursement expense | Ranks |
|------------------|-------------------------------|-------|
| John             | 100                           | 4     |
| Mathew           | 200                           | 3     |
| Tom              | 25                            | 6     |
| Mark             | 360                           | 2     |
| Lily             | 100                           | 4     |
| Marc             | 560                           | 1     |

As seen above, the RANK function gives duplicate numbers the same rank. However, the presence of duplicate numbers affects the ranks of subsequent numbers. For example, as shown above, in a list of integers sorted in ascending order, the number 100 appears twice with a rank of 4. The next value (25) will be ranked 6 (no number will be ranked 5).

If we want unique ranks, we can use the formula:

**`=RANK(C5,$C$5:C$10,0)+COUNTIF($C$5:C5,C5)-1`**

The screenshot shows the Excel formula bar with the formula `=RANK(C5,$C$5:C$10,0)+COUNTIF($C$5:C5,C5)-1`. Below it is a table titled "RANK Function" with the following data:

| Name of Employee | Medical reimbursement expense | Ranks | New Rank  |
|------------------|-------------------------------|-------|---|
| John             | 100                           | 4     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |
| Mathew           | 200                           | 3     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |
| Tom              | 25                            | 6     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |
| Mark             | 360                           | 2     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |
| Lily             | 100                           | 4     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |
| Marc             | 560                           | 1     | <code>=RANK(C5,\$C\$5:C\$10,0)+COUNTIF(\$C\$5:C5,C5)-1</code> |



We will get the results below:

|    | A | B                       | C                                    | D            | E               |
|----|---|-------------------------|--------------------------------------|--------------|-----------------|
| 1  |   |                         |                                      |              |                 |
| 2  |   | <b>RANK Function</b>    |                                      |              |                 |
| 3  |   |                         |                                      |              |                 |
| 4  |   | <b>Name of Employee</b> | <b>Medical reimbursement expense</b> | <b>Ranks</b> | <b>New Rank</b> |
| 5  |   | John                    | 100                                  | 4            | 4               |
| 6  |   | Mathew                  | 200                                  | 3            | 3               |
| 7  |   | Tom                     | 25                                   | 6            | 6               |
| 8  |   | Mark                    | 360                                  | 2            | 2               |
| 9  |   | Lily                    | 100                                  | 4            | 5               |
| 10 |   | Marc                    | 560                                  | 1            | 1               |
| 11 |   |                         |                                      |              |                 |

For ascending order, the formula would be:

**=RANK.EQ(C5,\$C\$5:C\$10,1)+COUNTIF(\$C\$5:C5,C5)-1**

In both formulas, it's the COUNTIF function that does the trick. We used COUNTIF to find out the number of times the number occurred that was ranked. In COUNTIF formula, the range consists of a single cell (\$C\$5:C5). As we locked only the first reference (\$C\$5), the last relative reference (C5) changes based on the row where the formula is copied. Thus, for row 7, the range expands to \$C\$5:C10, and the value in C10 is compared to each of the above cells.

Thus, for all unique values and first occurrences of duplicate values, COUNTIF returns 1; and we subtract 1 at the end of the formula to restore the original rank.

For ranks occurring the second time, COUNTIF returns 2. By subtracting 1, we increased the rank by 1 point, thus preventing duplicates. If there happen to be 3 plus occurrences of the same value, COUNTIF()-1 would add 2 to their ranking, and so on.

### Things to remember about the RANK Function

1. #N/A! error – Occurs when the given number is not present in the supplied reference. Also, the RANK function does not recognize text representations of numbers as numeric values, so we will also get the #N/A error if the values in the supplied ref array are text values.
2. If we provide logical values, we will get the #VALUE! error.

[Click here to download the sample Excel file](#)

**SMALL**

Get the nth smallest value

array

k

**What is the SMALL Function?**

The SMALL Function is categorized as a **Statistical function**. It will return numeric values based on their position in a list ranked by value. Essentially, it calculates and returns the k smallest value in a dataset.

In **financial analysis**, the SMALL function can be useful in finding the smallest value in a given set of values. For example, if all salespersons are given the same target, we can find out which one achieved the sales target in the shortest time for a given year.

**Formula****=SMALL(array,k)**

The SMALL function uses the following arguments:

1. **Array** (required argument) – It is an array or range of numerical data for which we want to find the k smallest value.
2. **K** (required argument) – It is the smallest value from the given array.

**How to use the SMALL Function in Excel?**

To understand the uses of the SMALL function, let us consider a few examples:

## Example

Suppose we are given the following data about several runners, along with their start and finish times:

|   | A | B                     | C            | D             | E           |
|---|---|-----------------------|--------------|---------------|-------------|
| 1 |   |                       |              |               |             |
| 2 |   | <b>SMALL Function</b> |              |               |             |
| 3 |   |                       |              |               |             |
| 4 |   | <b>Name</b>           | <b>Start</b> | <b>Finish</b> | <b>Time</b> |
| 5 |   | Sean                  | 7:00:00 AM   | 8:10:05 AM    | 1:10:05 AM  |
| 6 |   | John                  | 7:00:10 AM   | 8:10:45 AM    | 1:10:35 AM  |
| 7 |   | Justin                | 7:00:15 AM   | 8:10:25 AM    | 1:10:10 AM  |
| 8 |   | Linda                 | 7:00:01 AM   | 8:12:05 AM    | 1:12:04 AM  |
| 9 |   | William               | 7:00:05 AM   | 8:15:05 AM    | 1:15:00 AM  |

We can use the SMALL function to retrieve the nth lowest value from a set of data, that is, the first, second, or third fastest times in a race.

The formula to be used in finding the winner will be:

|    | A | B                     | C                      | D             | E           | F |
|----|---|-----------------------|------------------------|---------------|-------------|---|
| 1  |   |                       |                        |               |             |   |
| 2  |   | <b>SMALL Function</b> |                        |               |             |   |
| 3  |   |                       |                        |               |             |   |
| 4  |   | <b>Name</b>           | <b>Start</b>           | <b>Finish</b> | <b>Time</b> |   |
| 5  |   | Sean                  | 7:00:00 AM             | 8:10:05 AM    | 1:10:05 AM  |   |
| 6  |   | John                  | 7:00:10 AM             | 8:10:45 AM    | 1:10:35 AM  |   |
| 7  |   | Justin                | 7:00:15 AM             | 8:10:25 AM    | 1:10:10 AM  |   |
| 8  |   | Linda                 | 7:00:01 AM             | 8:12:05 AM    | 1:12:04 AM  |   |
| 9  |   | William               | 7:00:05 AM             | 8:15:05 AM    | 1:15:00 AM  |   |
| 10 |   |                       |                        |               |             |   |
| 11 |   | <b>Winner</b>         | <b>=SMALL(E5:E9,1)</b> |               |             |   |

The formula to be used in finding the first runner-up will be:

The screenshot shows the Excel formula bar with the formula `=SMALL(E5:E9,2)` entered. Below the formula bar, a table titled "SMALL Function" displays the following data:

| Name    | Start      | Finish     | Time       |
|---------|------------|------------|------------|
| Sean    | 7:00:00 AM | 8:10:05 AM | 1:10:05 AM |
| John    | 7:00:10 AM | 8:10:45 AM | 1:10:35 AM |
| Justin  | 7:00:15 AM | 8:10:25 AM | 1:10:10 AM |
| Linda   | 7:00:01 AM | 8:12:05 AM | 1:12:04 AM |
| William | 7:00:05 AM | 8:15:05 AM | 1:15:00 AM |

Below the table, the results are shown:

|               |                 |
|---------------|-----------------|
| Winner        | 1:10:05 AM      |
| 1st Runner-up | =SMALL(E5:E9,2) |

The formula to be used in finding the second runner-up will be:

The screenshot shows the Excel formula bar with the formula `=SMALL(E5:E9,3)` entered. Below the formula bar, a table titled "SMALL Function" displays the following data:

| Name    | Start      | Finish     | Time       |
|---------|------------|------------|------------|
| Sean    | 7:00:00 AM | 8:10:05 AM | 1:10:05 AM |
| John    | 7:00:10 AM | 8:10:45 AM | 1:10:35 AM |
| Justin  | 7:00:15 AM | 8:10:25 AM | 1:10:10 AM |
| Linda   | 7:00:01 AM | 8:12:05 AM | 1:12:04 AM |
| William | 7:00:05 AM | 8:15:05 AM | 1:15:00 AM |

Below the table, the results are shown:

|               |                 |
|---------------|-----------------|
| Winner        | 1:10:05 AM      |
| 1st Runner-up | 1:10:10 AM      |
| 2nd Runner up | =SMALL(E5:E9,3) |

We get the following results:

The screenshot shows the final results of the SMALL function in cell C13. The formula bar shows `=SMALL(E5:E9,3)`. Below the formula bar, a table titled "SMALL Function" displays the following data:

| Name    | Start      | Finish     | Time       |
|---------|------------|------------|------------|
| Sean    | 7:00:00 AM | 8:10:05 AM | 1:10:05 AM |
| John    | 7:00:10 AM | 8:10:45 AM | 1:10:35 AM |
| Justin  | 7:00:15 AM | 8:10:25 AM | 1:10:10 AM |
| Linda   | 7:00:01 AM | 8:12:05 AM | 1:12:04 AM |
| William | 7:00:05 AM | 8:15:05 AM | 1:15:00 AM |

Below the table, the results are shown:

|               |            |
|---------------|------------|
| Winner        | 1:10:05 AM |
| 1st Runner-up | 1:10:10 AM |
| 2nd Runner up | 1:10:35 AM |

### Things to remember about the SMALL Function

1. #NUM! error – Occurs when:
  - The given array is empty; or
  - The given value of k is less than 1 or greater than the number of values in the given array.
2. #VALUE! error – Occurs if the given k is non-numeric.
3. If n is the number of data points in an array, SMALL(array,1) equals the smallest value, and SMALL(array,n) equals the largest value.
4. Sometimes, the SMALL function will return a wrong value, or will return a #NUM! error, even though the supplied value of k is between 1 and the number of values in the supplied array. It will occur when we provide text values or text representations of numbers within the supplied array. The function ignores text values. Therefore, this problem may arise if the values in the supplied array are text representations of numbers, instead of actual values. We can solve this issue by converting all array values into numeric values.

[Click here to download the sample Excel file](#)

**For more information or  
to register for our courses  
please contact us at:**

[learning@corporatefinanceinstitute.com](mailto:learning@corporatefinanceinstitute.com)

[www.corporatefinanceinstitute.com](http://www.corporatefinanceinstitute.com)

For more eBooks please visit:

[corporatefinanceinstitute.com/resources/ebooks/](http://corporatefinanceinstitute.com/resources/ebooks/)