
## Technical Institute of Administration

### Business Administration

### Computer Skills

# 3. Microsoft Excel - Functions

Lecturer:

Sipan M. Hameed

https://www.sipan.dev/

2025-2026

Table of Contents

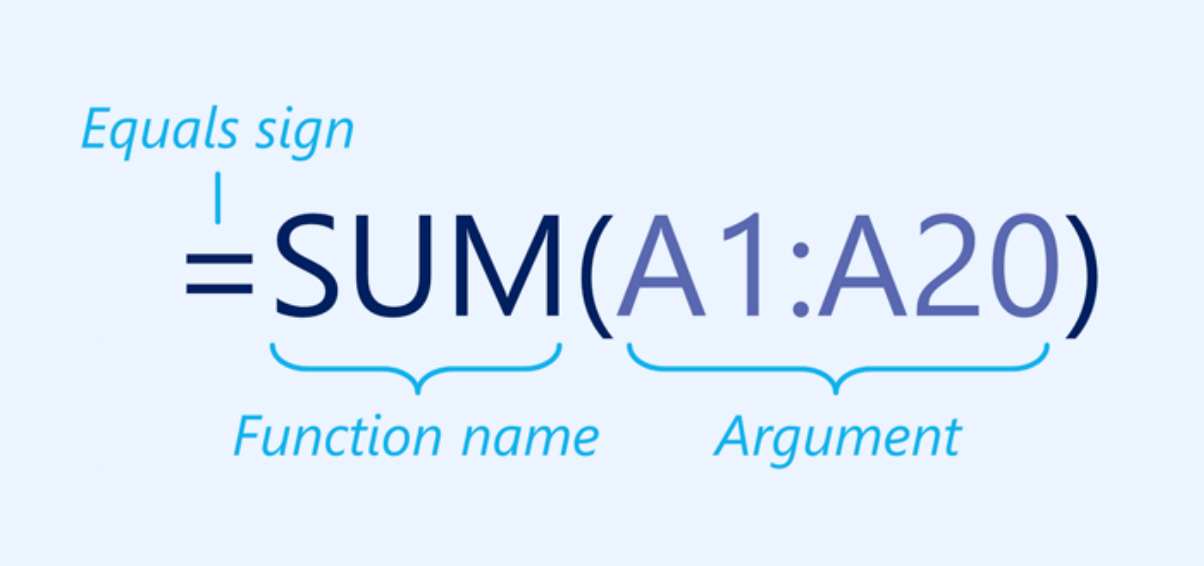
# 3. Functions

## 3.1. Introduction

A **function** is a **predefined formula** that performs calculations using specific values in a particular order. Excel includes many common functions that can be used to quickly find the **sum**, **average**, **count**, **maximum value**, and **minimum value** for a range of cells. In order to use functions correctly, you'll need to understand the different **parts of a function** and how to create **arguments** to calculate values and cell references.

### 3.1.1. The parts of a function

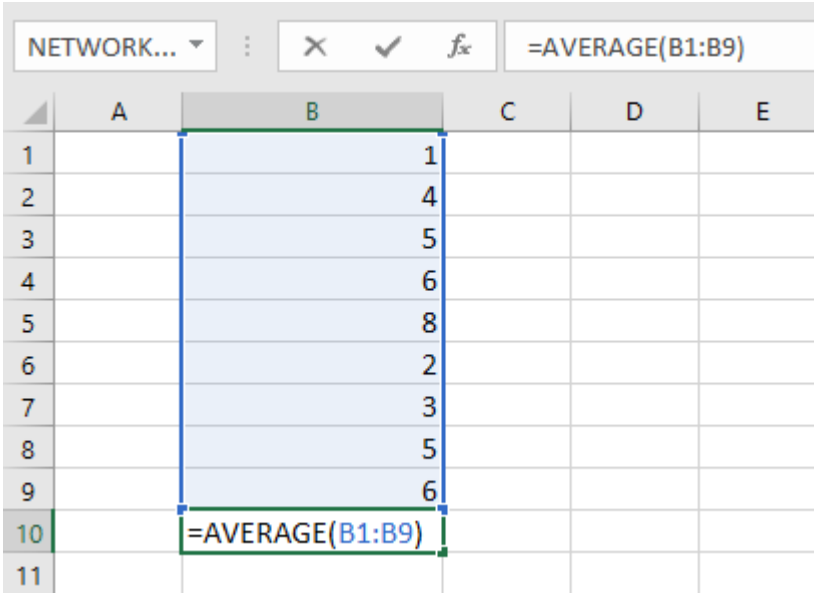
In order to work correctly, a function must be written a specific way, which is called the **syntax**. The basic syntax for a function is the **equals sign (=)**, the **function name** (SUM, for example), and one or more **arguments**. Arguments contain the information you want to calculate. The function in the example below would add the values of the cell range A1:A20.



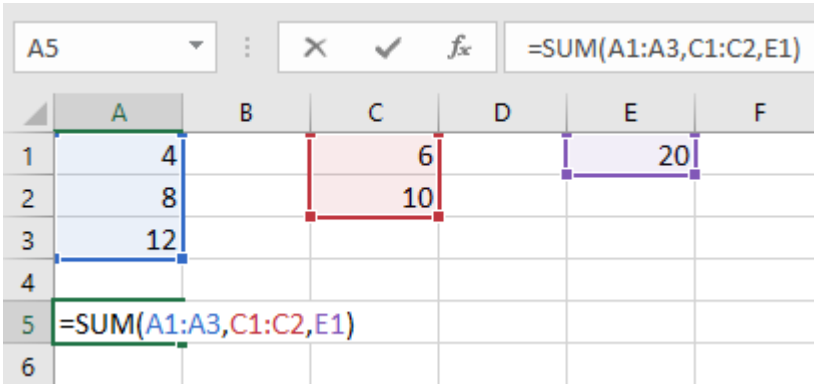

### 3.1.2. Working with arguments

Arguments can refer to both **individual cells** and **cell ranges** and must be enclosed within **parentheses**. You can include one argument or multiple arguments, depending on the syntax required for the function.

For example, the function **=AVERAGE(B1:B9)** would calculate the **average** of the values in the cell range B1:B9. This function contains only one argument.




Multiple arguments must be separated by a **comma**. For example, the function **=SUM(A1:A3, C1:C2, E1)** will **add** the values of all of the cells in the three arguments.




## 3.2. Creating a function

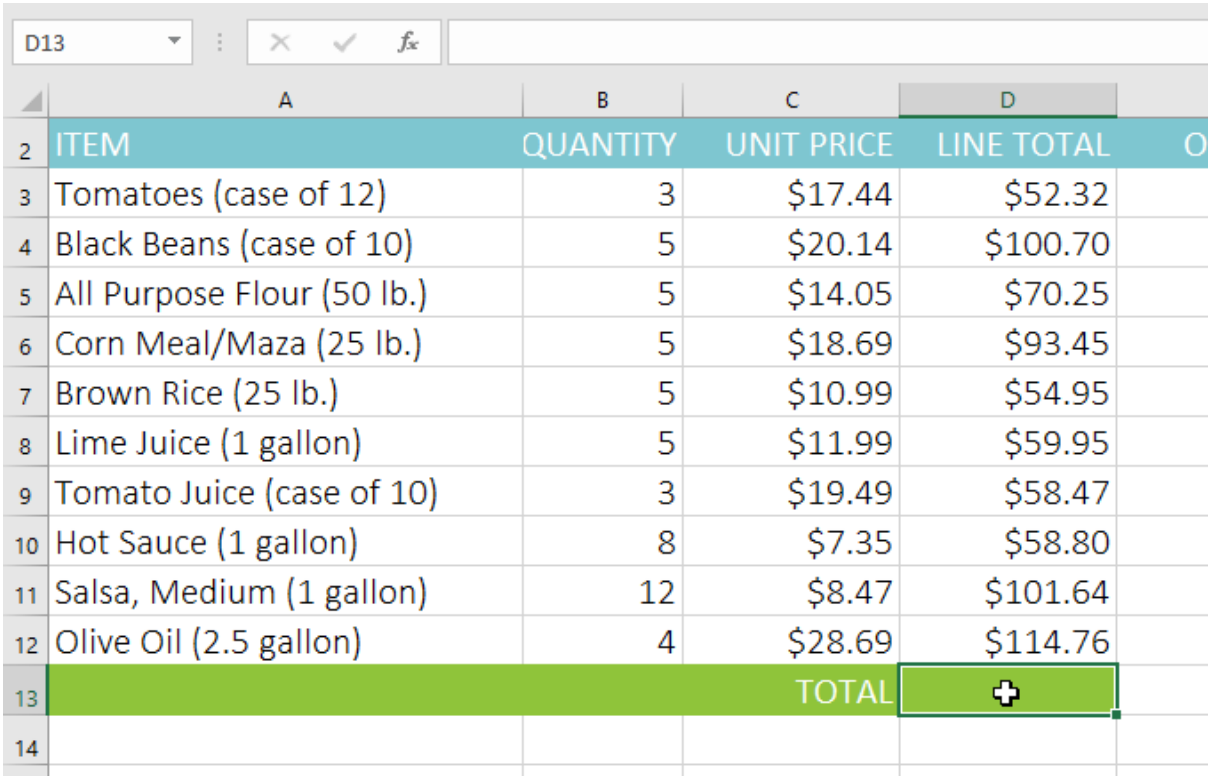
There are a variety of functions available in Excel. Here are some of the most common functions you'll use:

- **SUM**: This function **adds** all of the values of the cells in the argument.
- **AVERAGE**: This function determines the **average** of the values included in the argument. It calculates the sum of the cells and then divides that value by the number of cells in the argument.
- **COUNT**: This function **counts** the number of cells with numerical data in the argument. This function is useful for quickly counting items in a cell range.
- **MAX**: This function determines the **highest cell value** included in the argument.
- **MIN**: This function determines the **lowest cell value** included in the argument.

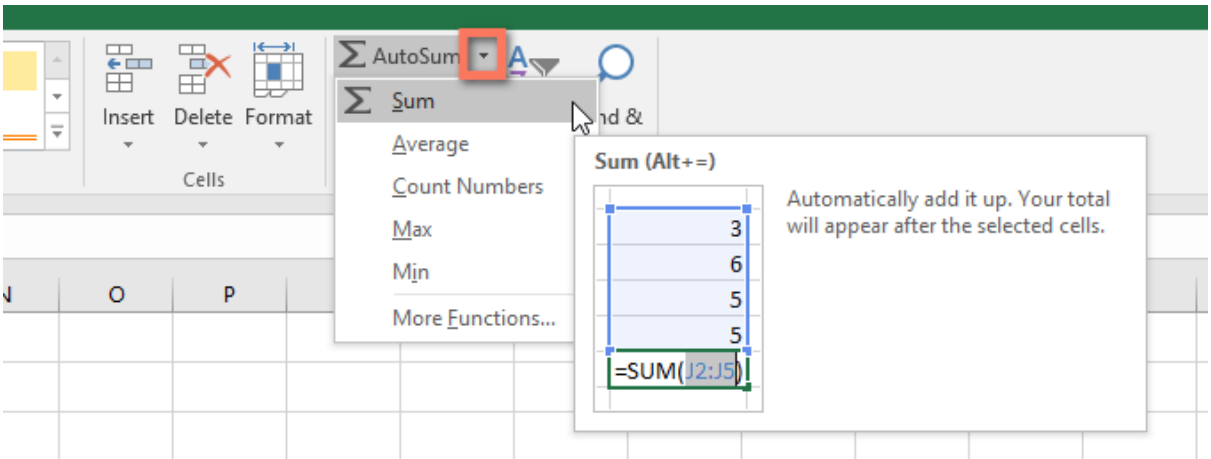
### 3.2.1. To create a function using the AutoSum command:

The **AutoSum** command allows you to automatically insert the most common functions into your formula, including SUM, AVERAGE, COUNT, MIN, and MAX. In the example below, we'll use the **SUM** function to calculate the **total cost** for a list of recently ordered items.

1. Select the **cell** that will contain the function. In our example, we'll select cell **D13**.

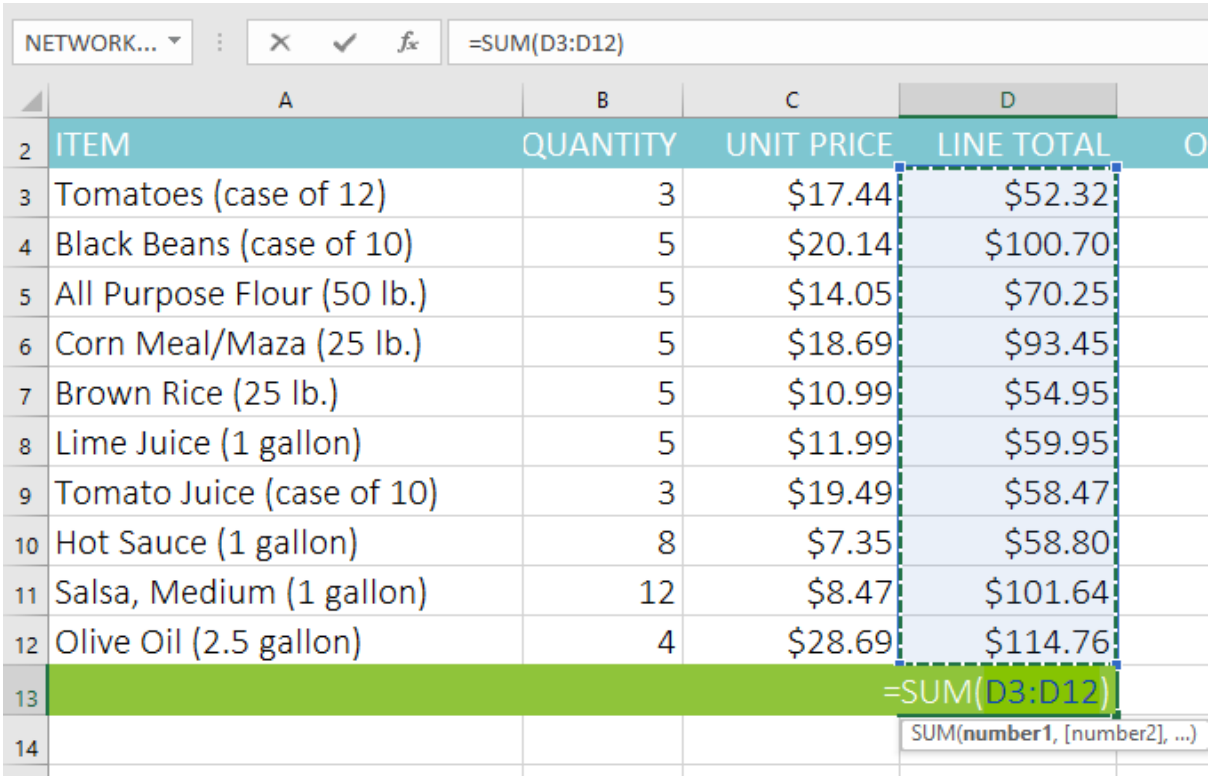



2. In the **Editing** group on the **Home** tab, click the **arrow** next to the **AutoSum** command. Next, choose the **desired function** from the drop-down menu. In our example, we'll select **Sum**.

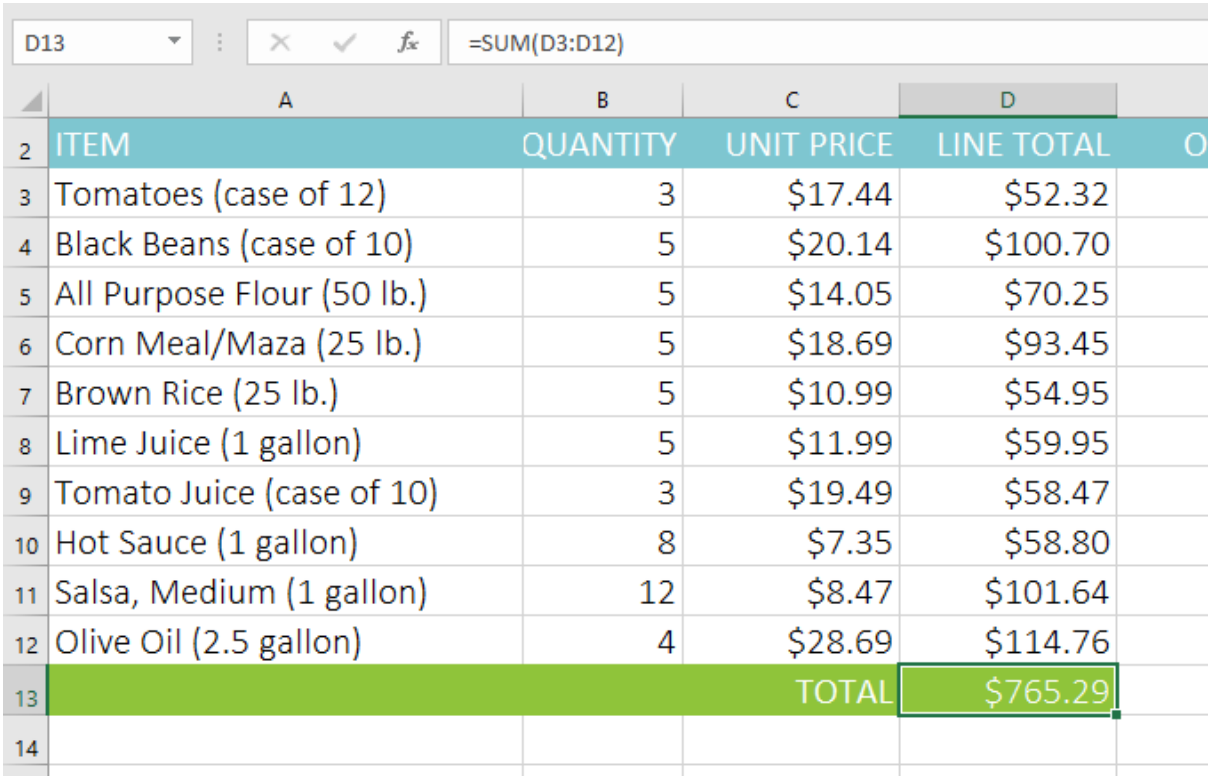



3. Excel will place the **function** in the cell and automatically select a **cell range** for the argument. In our example, cells **D3:D12** were selected automatically; their values will

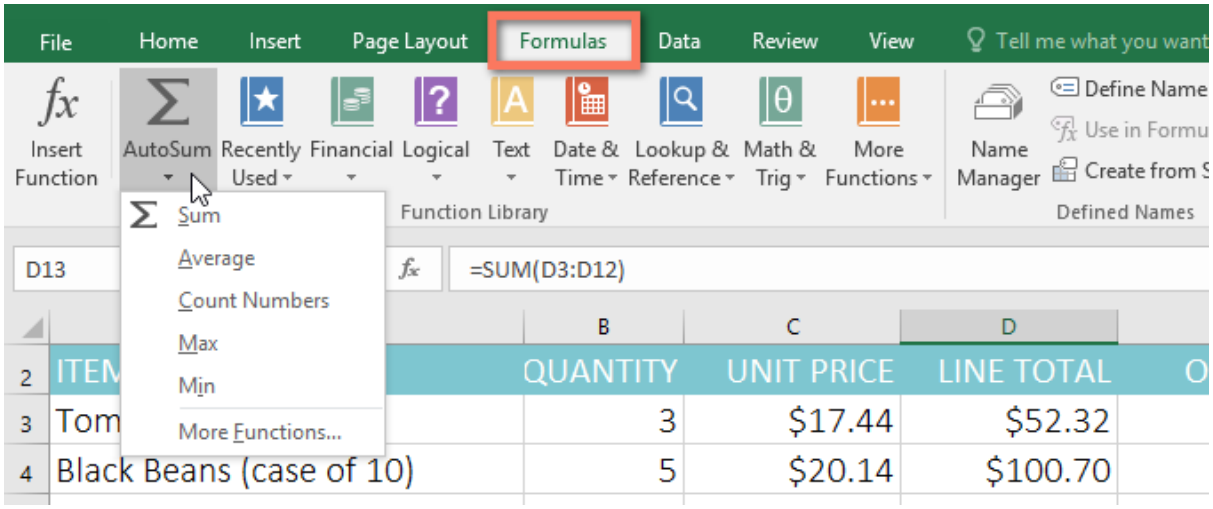
be **added** to calculate the total cost. If Excel selects the wrong cell range, you can manually enter the desired cells into the argument.




4. Press **Enter** on your keyboard. The function will be **calculated**, and the **result** will appear in the cell. In our example, the sum of D3:D12 is **$765.29**.



The **AutoSum** command can also be accessed from the **Formulas** tab on the **Ribbon**.



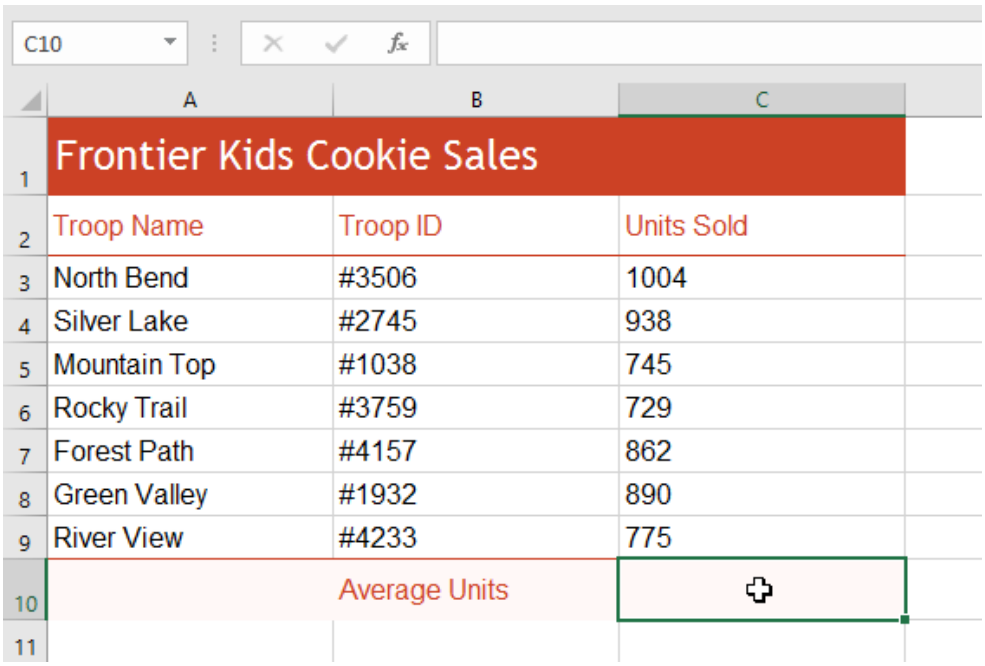

You can also use the **Alt+=** keyboard shortcut instead of the AutoSum command. To use this shortcut, hold down the **Alt** key and then press the **equals sign**.

Watch the video below to see this shortcut in action.

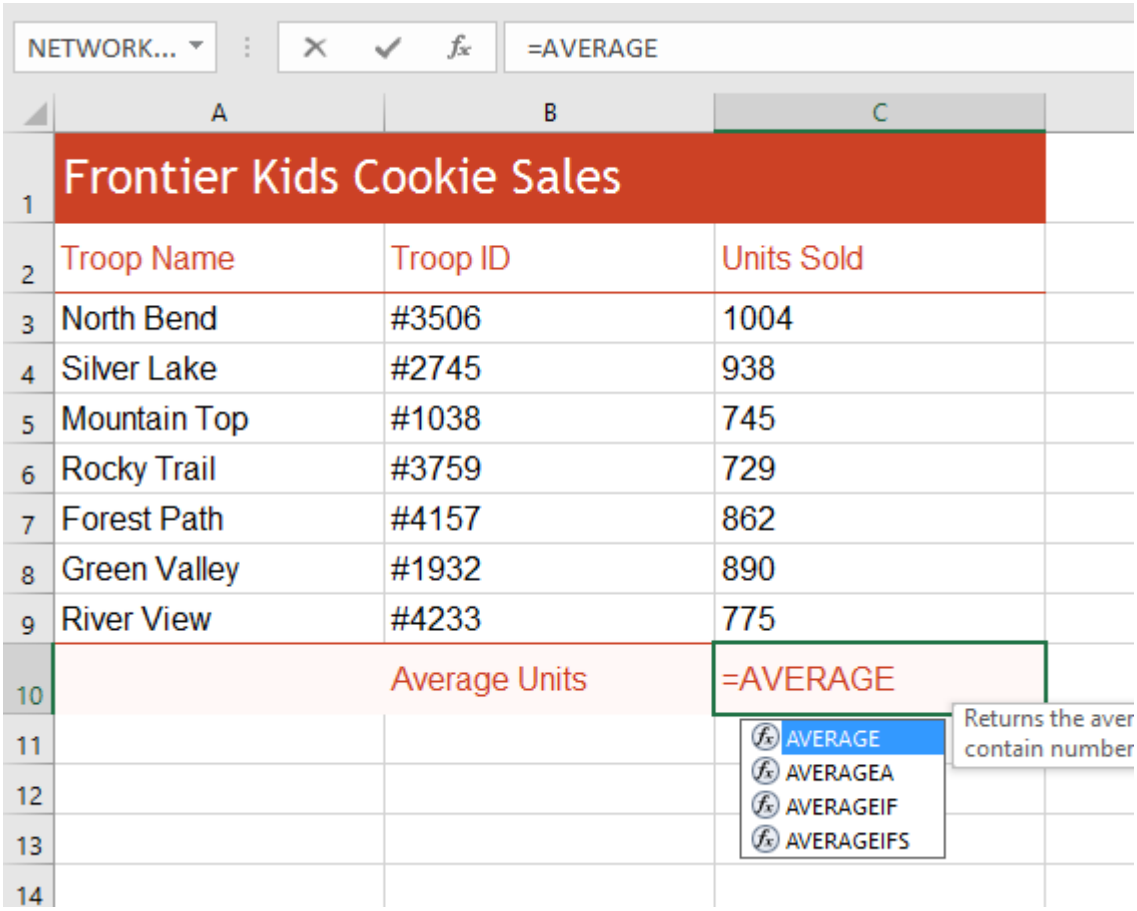
## 3.3. To enter a function manually:

If you already know the function name, you can easily type it yourself. In the example below (a tally of cookie sales), we'll use the **AVERAGE** function to calculate the **average number of units sold** by each troop.

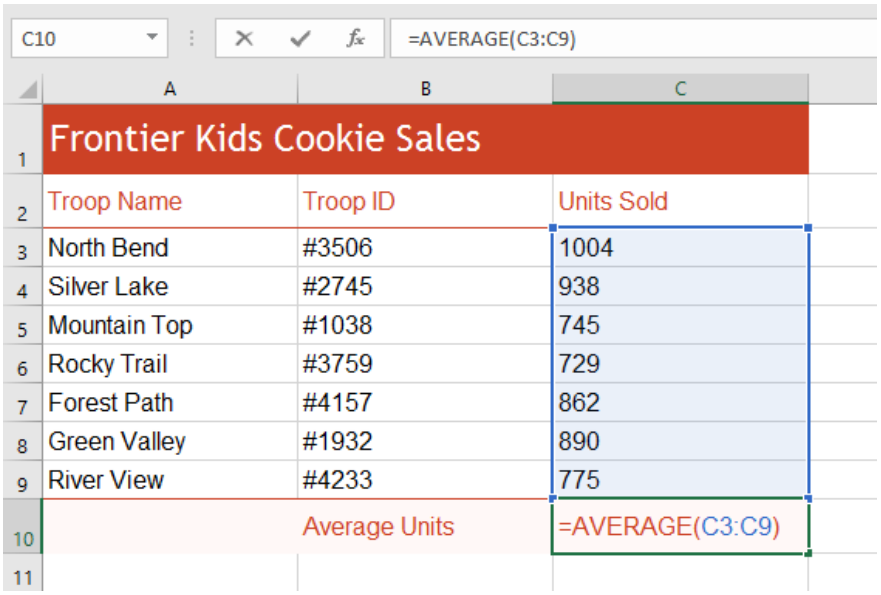
1. Select the **cell** that will contain the function. In our example, we'll select cell **C10**.



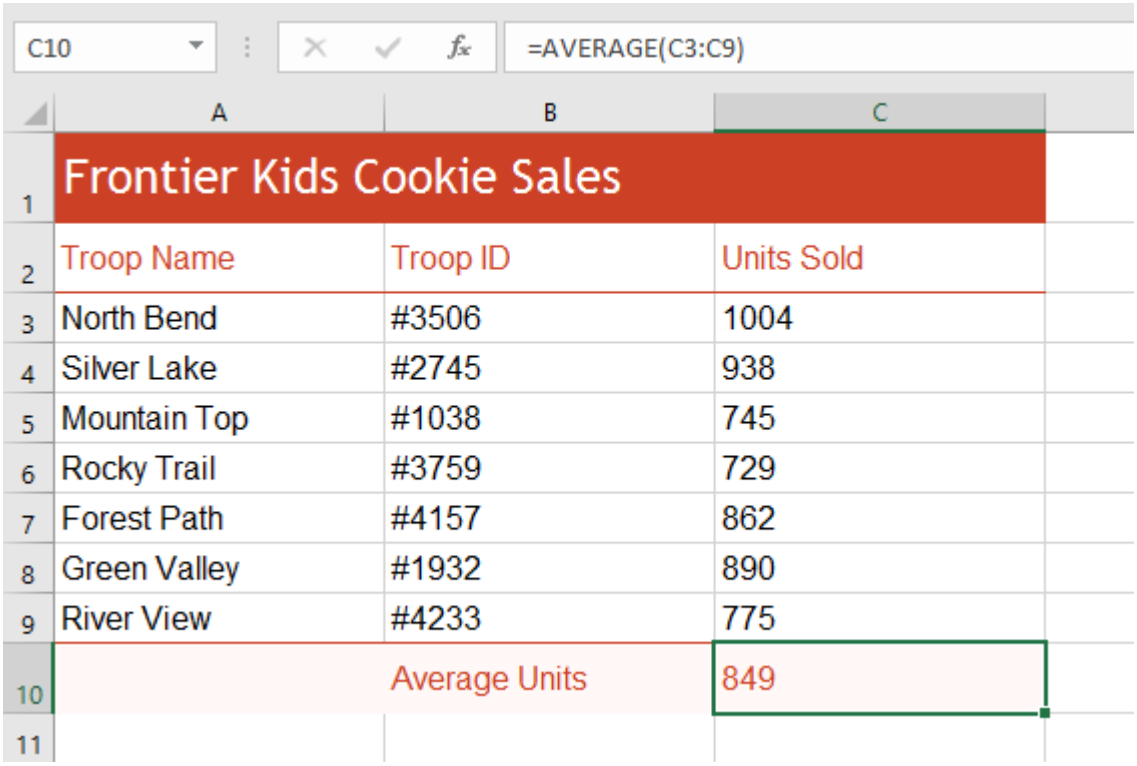
2. Type the **equals sign (=),** and enter the desired **function name**. You can also select the desired function from the list of **suggested functions** that appears below the cell as you type. In our example, we'll type **=AVERAGE**.




3. Enter the **cell range** for the argument inside **parentheses**. In our example, we'll type **(C3:C9)**. This formula will add the values of cells C3:C9, then divide that value by the total number of values in the range.



4. Press **Enter** on your keyboard. The function will be calculated, and the **result** will appear in the cell. In our example, the average number of units sold by each troop is **849**.



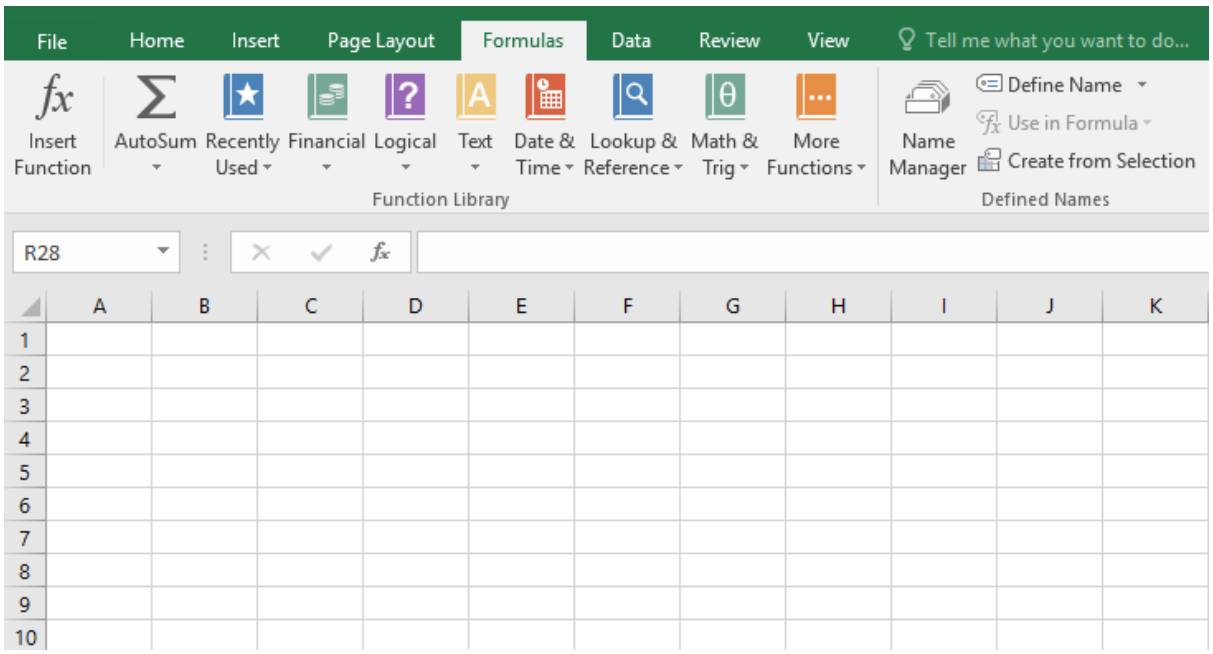

Excel **will not always tell you** if your formula contains an error, so it's up to you to check all of your formulas.

## 3.4. The Function Library

While there are hundreds of functions in Excel, the ones you'll use the most will depend on the **type of data** your workbooks contain. There's no need to learn every single function, but exploring some of the different **types** of functions will help you as you create new projects. You can even use the **Function Library** on the **Formulas** tab to browse functions by category, such as **Financial**, **Logical**, **Text**, and **Date & Time**.

To access the **Function Library**, select the **Formulas** tab on the **Ribbon**. Look for the **Function Library** group.

Click the buttons in the interactive below to learn more about the different types of functions in Excel.



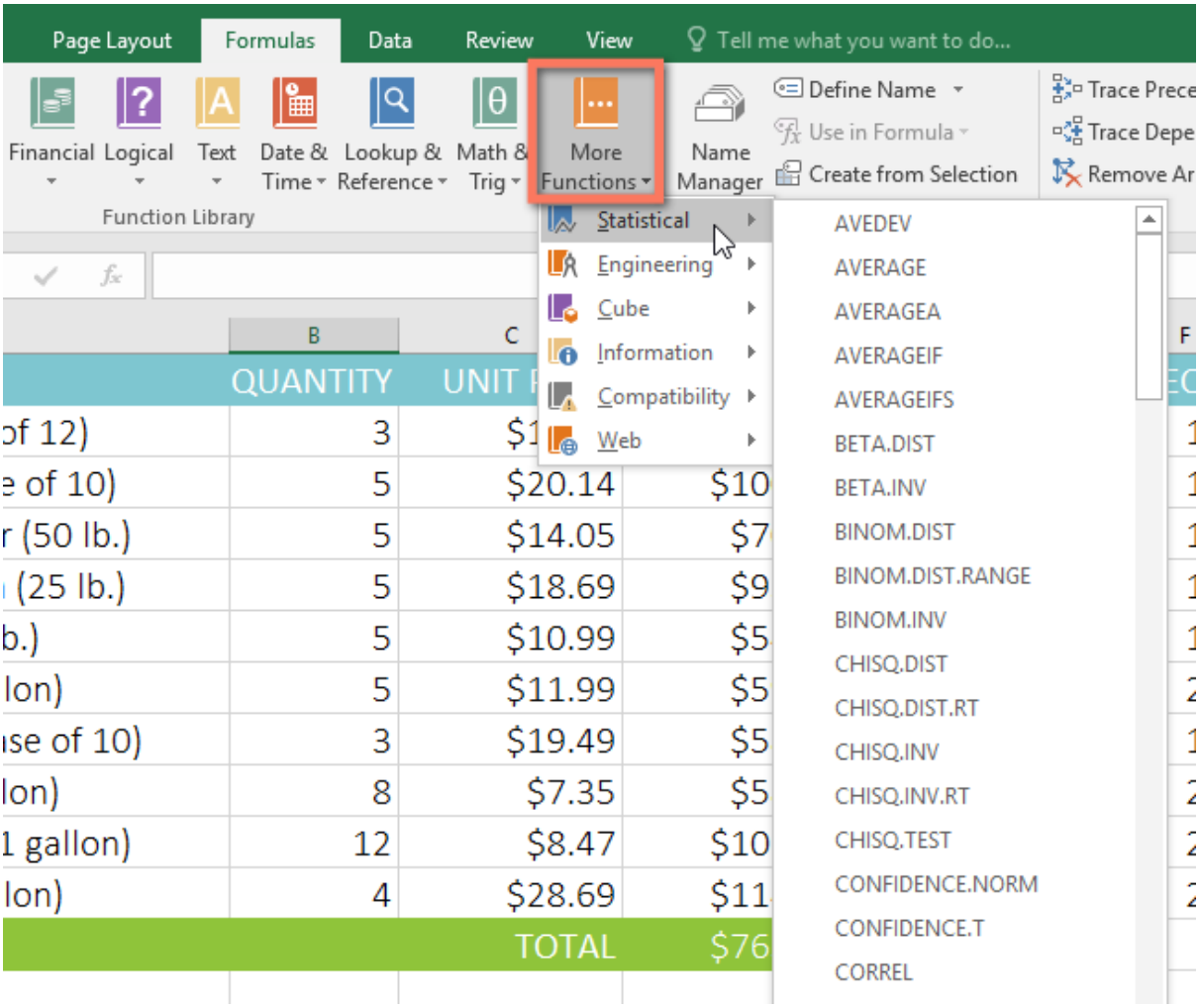

### 3.4.1. To insert a function from the Function Library:

In the example below, we'll use the COUNTA function to count the total number of items in the **Items** column. Unlike COUNT, **COUNTA** can be used to tally cells that contain data of any kind, not just numerical data.

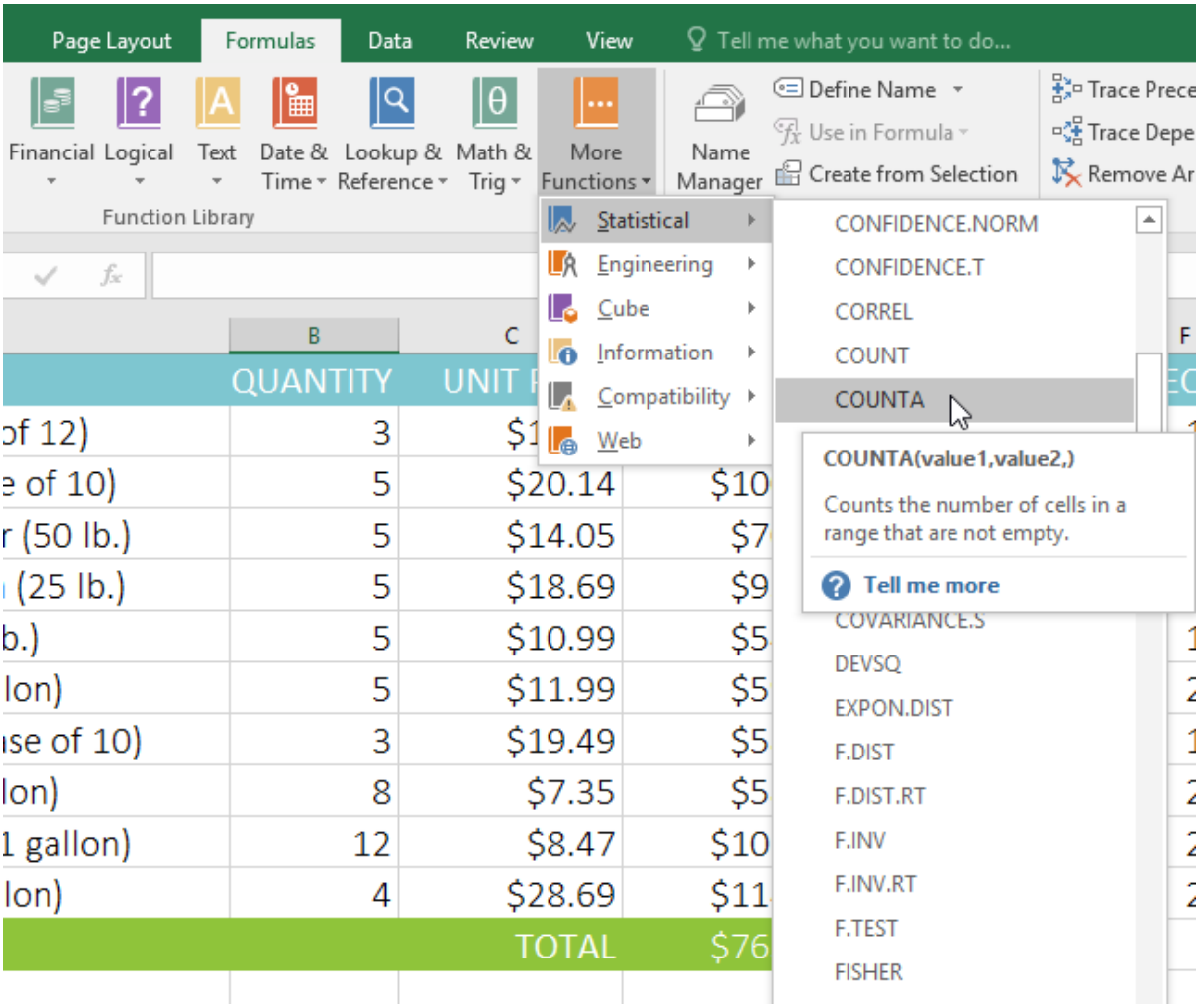
1. Select the **cell** that will contain the function. In our example, we'll select cell **B17**.

| | A | B | C | D |
|---|---|---|---|---|
| 2 | ITEM | QUANTITY | UNIT PRICE | LINE TOTAL |
| 3 | Tomatoes (case of 12) | 3 | $17.44 | $52.32 |
| 4 | Black Beans (case of 10) | 5 | $20.14 | $100.70 |
| 5 | All Purpose Flour (50 lb.) | 5 | $14.05 | $70.25 |
| 6 | Corn Meal/Maza (25 lb.) | 5 | $18.69 | $93.45 |
| 7 | Brown Rice (25 lb.) | 5 | $10.99 | $54.95 |
| 8 | Lime Juice (1 gallon) | 5 | $11.99 | $59.95 |
| 9 | Tomato Juice (case of 10) | 3 | $19.49 | $58.47 |
| 10 | Hot Sauce (1 gallon) | 8 | $7.35 | $58.80 |
| 11 | Salsa, Medium (1 gallon) | 12 | $8.47 | $101.64 |
| 12 | Olive Oil (2.5 gallon) | 4 | $28.69 | $114.76 |
| 13 | | | TOTAL | $765.29 |
| 14 | | | | |
| 15 | | | | |
| 16 | PURCHASE ORDER SUMMARY | | | |
| 17 | Total items ordered | | | |
| 18 | Most expensive item | | | |
| 19 | Average days in transit | | | |
| 20 | | | | |

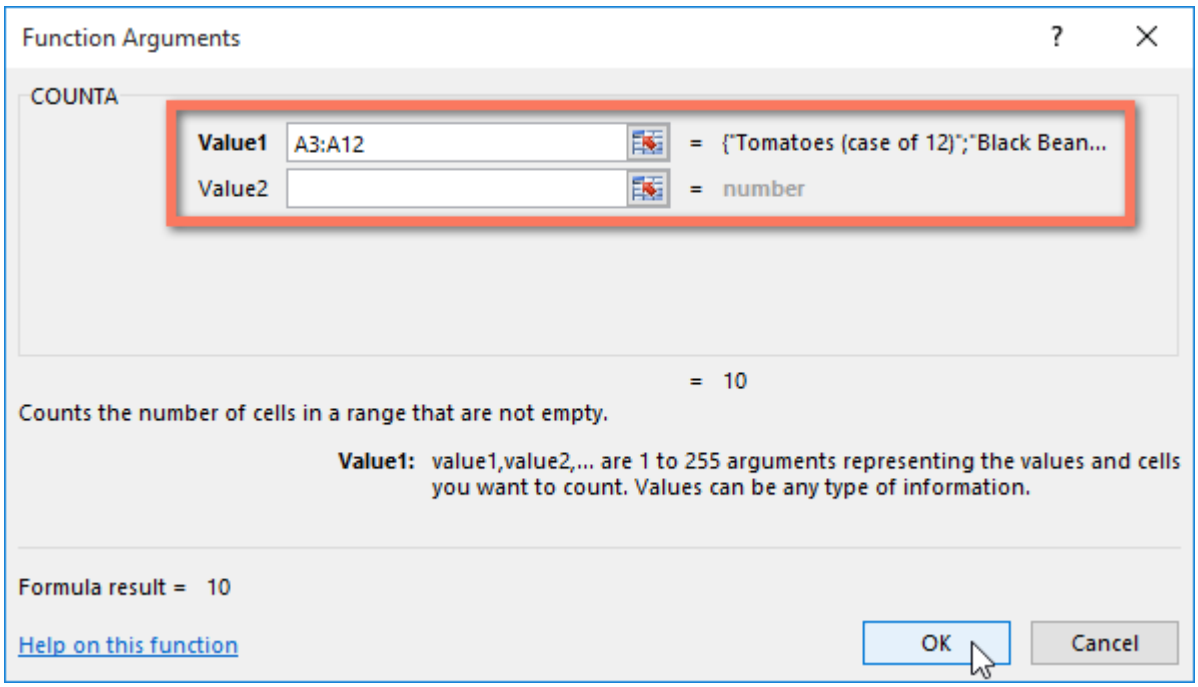
2. Click the **Formulas** tab on the **Ribbon** to access the **Function Library**.
3. From the **Function Library** group, select the desired **function category**. In our example, we'll choose **More Functions**, then hover the mouse over **Statistical**.




4. Select the **desired function** from the drop-down menu. In our example, we'll select the **COUNTA** function, which will count the number of cells in the **Items** column that are not empty.




5. The **Function Arguments** dialog box will appear. Select the **Value1** field, then enter or select the desired cells. In our example, we'll enter the cell range **A3:A12**. You may continue to add arguments in the **Value2** field, but in this case we only want to count the number of cells in the cell range **A3:A12**.
6. When you're satisfied, click **OK**.




7. The function will be **calculated**, and the **result** will appear in the cell. In our example, the result shows that a total of **10 items** were ordered.


| | A | B | C | D |
|---|---|---|---|---|
| 2 | ITEM | QUANTITY | UNIT PRICE | LINE TOTAL |
| 3 | Tomatoes (case of 12) | 3 | $17.44 | $52.32 |
| 4 | Black Beans (case of 10) | 5 | $20.14 | $100.70 |
| 5 | All Purpose Flour (50 lb.) | 5 | $14.05 | $70.25 |
| 6 | Corn Meal/Maza (25 lb.) | 5 | $18.69 | $93.45 |
| 7 | Brown Rice (25 lb.) | 5 | $10.99 | $54.95 |
| 8 | Lime Juice (1 gallon) | 5 | $11.99 | $59.95 |
| 9 | Tomato Juice (case of 10) | 3 | $19.49 | $58.47 |
| 10 | Hot Sauce (1 gallon) | 8 | $7.35 | $58.80 |
| 11 | Salsa, Medium (1 gallon) | 12 | $8.47 | $101.64 |
| 12 | Olive Oil (2.5 gallon) | 4 | $28.69 | $114.76 |
| 13 | | | TOTAL | $765.29 |
| 14 | | | | |
| 15 | | | | |
| 16 | PURCHASE ORDER SUMMARY | | | |
| 17 | Total items ordered | 10 | | |
| 18 | Most expensive item | | | |
| 19 | Average days in transit | | | |
| 20 | | | | |

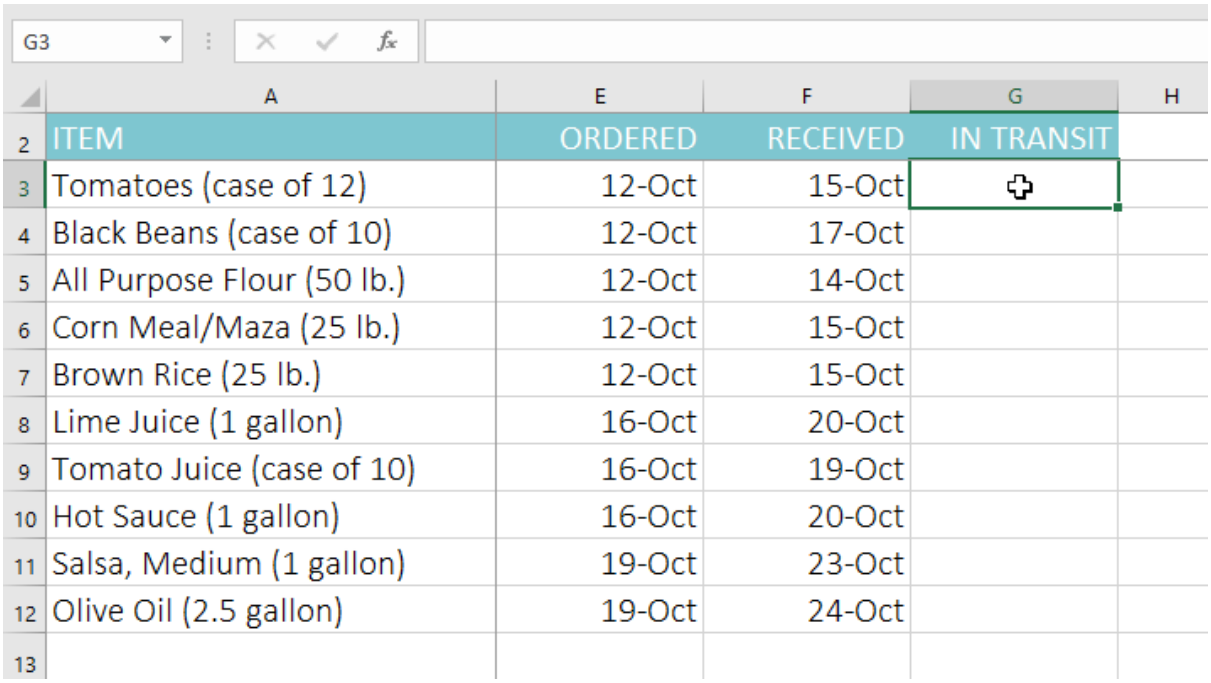
## 3.5. The Insert Function command

While the Function Library is a great place to browse for functions, sometimes you may prefer to **search** for one instead. You can do so using the **Insert Function** command. It may take some trial and error depending on the type of function you're looking for; however, with practice, the Insert Function command can be a powerful way to find a function quickly.

### 3.5.1. To use the Insert Function command:

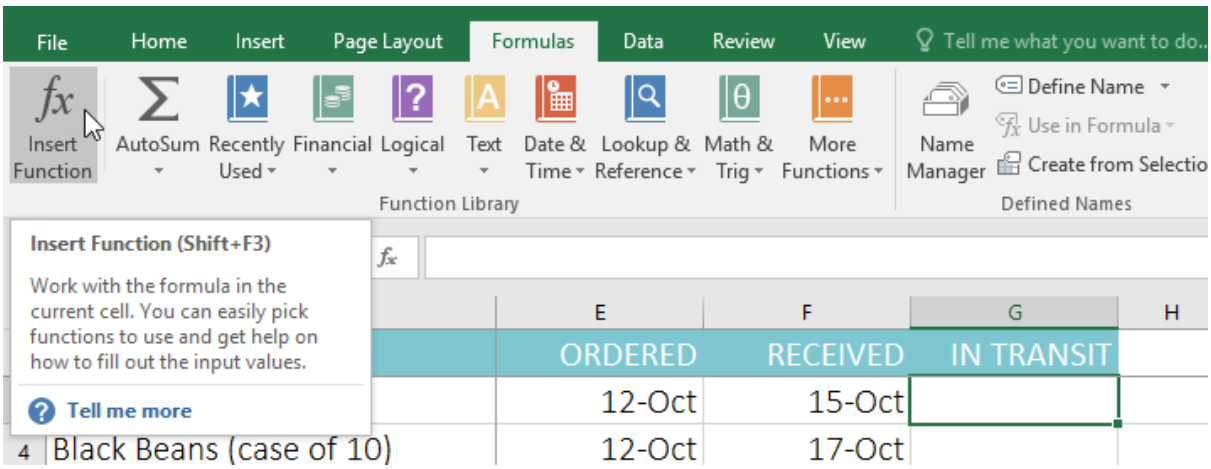
In the example below, we want to find a function that will calculate the **number of business days** it took to receive items after they were ordered. We'll use the dates in columns **E** and **F** to calculate the delivery time in column **G**.

1. Select the **cell** that will contain the function. In our example, we'll select cell **G3**.

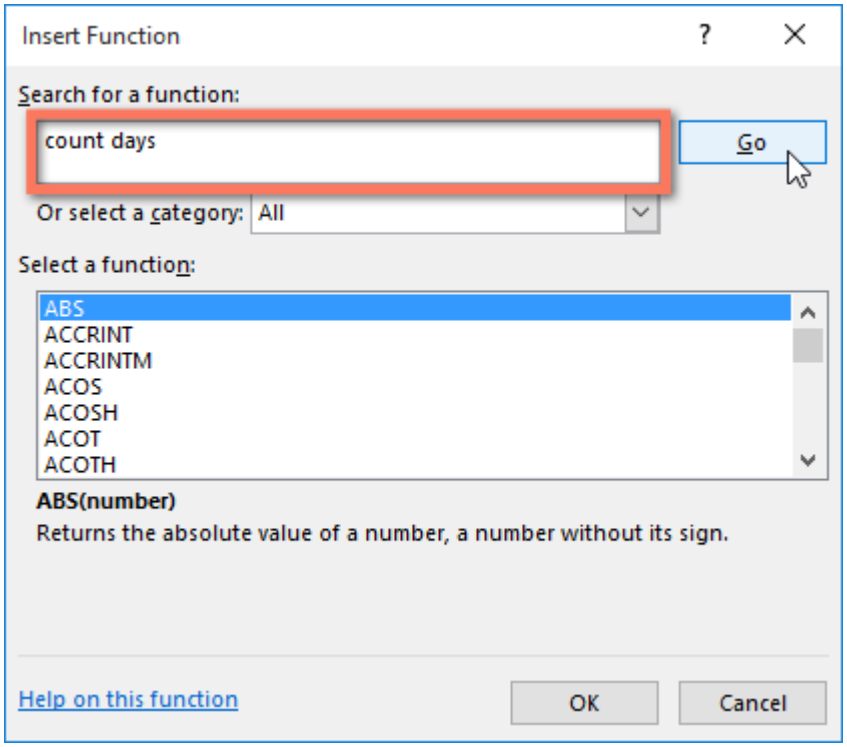


| | A | E | F | G | H |
|---|---|---|---|---|---|
| 2 | ITEM | ORDERED | RECEIVED | IN TRANSIT | |
| 3 | Tomatoes (case of 12) | 12-Oct | 15-Oct | | |
| 4 | Black Beans (case of 10) | 12-Oct | 17-Oct | | |
| 5 | All Purpose Flour (50 lb.) | 12-Oct | 14-Oct | | |
| 6 | Corn Meal/Maza (25 lb.) | 12-Oct | 15-Oct | | |
| 7 | Brown Rice (25 lb.) | 12-Oct | 15-Oct | | |
| 8 | Lime Juice (1 gallon) | 16-Oct | 20-Oct | | |
| 9 | Tomato Juice (case of 10) | 16-Oct | 19-Oct | | |
| 10 | Hot Sauce (1 gallon) | 16-Oct | 20-Oct | | |
| 11 | Salsa, Medium (1 gallon) | 19-Oct | 23-Oct | | |
| 12 | Olive Oil (2.5 gallon) | 19-Oct | 24-Oct | | |
| 13 | | | | | |

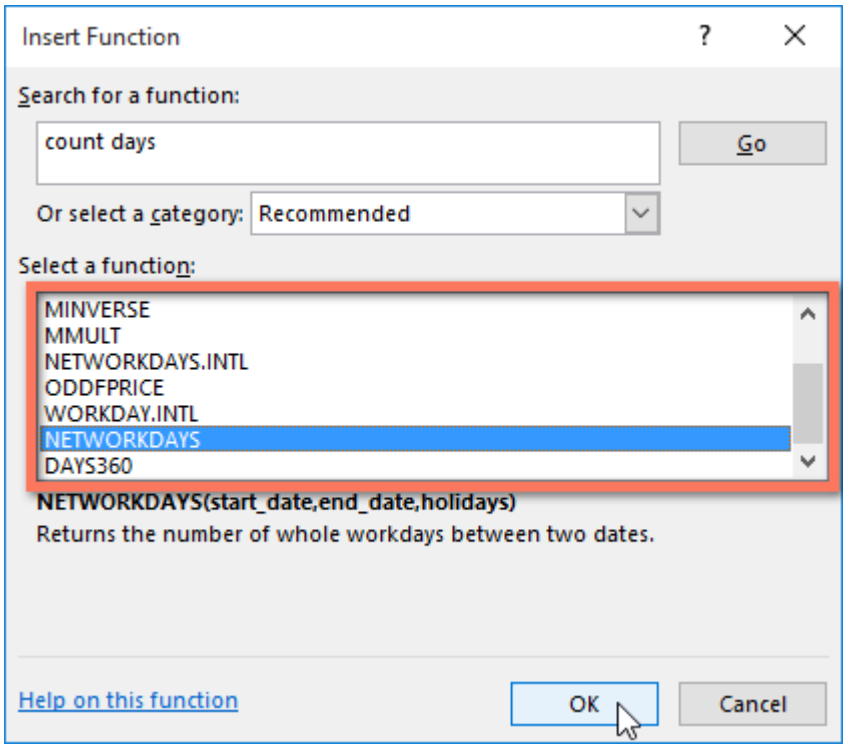
2. Click the **Formulas** tab on the **Ribbon**, then click the **Insert Function** command.



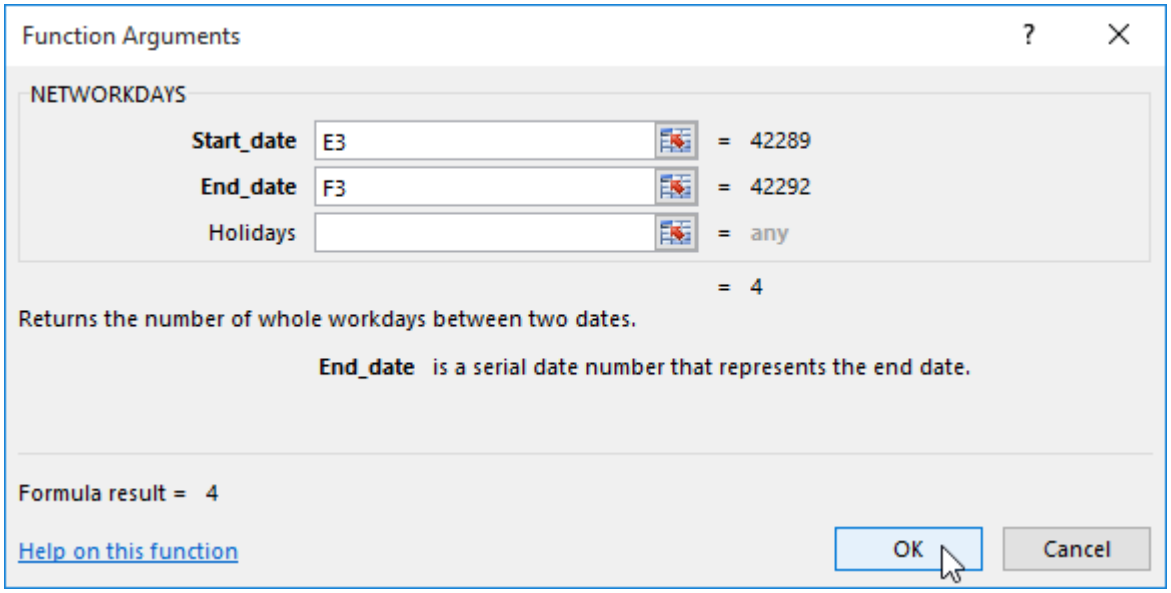
3. The **Insert Function** dialog box will appear.
4. Type a few **keywords** describing the calculation you want the function to perform, then click **Go**. In our example, we'll type **count days**, but you can also search by selecting a **category** from the drop-down list.



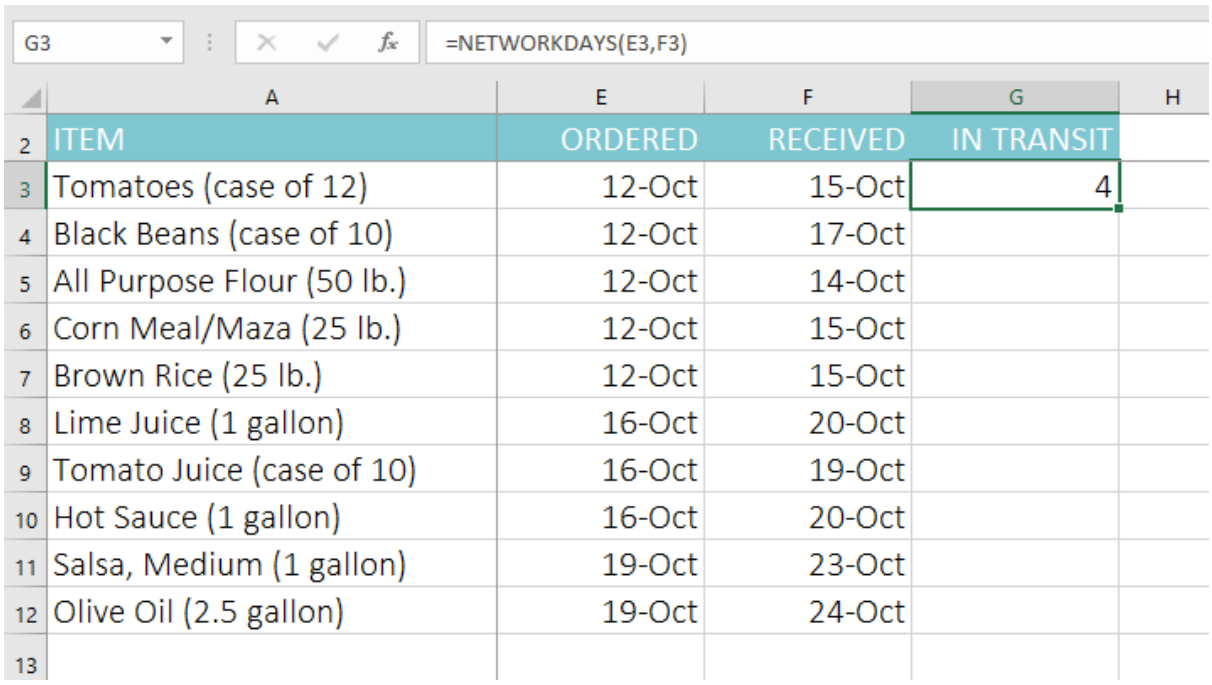

5. Review the **results** to find the desired function, then click **OK**. In our example, we'll choose **NETWORKDAYS**, which will count the number of business days between the ordered date and received date.



6. The **Function Arguments** dialog box will appear. From here, you'll be able to enter or select the cells that will make up the arguments in the function. In our example, we'll enter **E3** in the **Start_date** field and **F3** in the **End_date** field.
7. When you're satisfied, click **OK**.

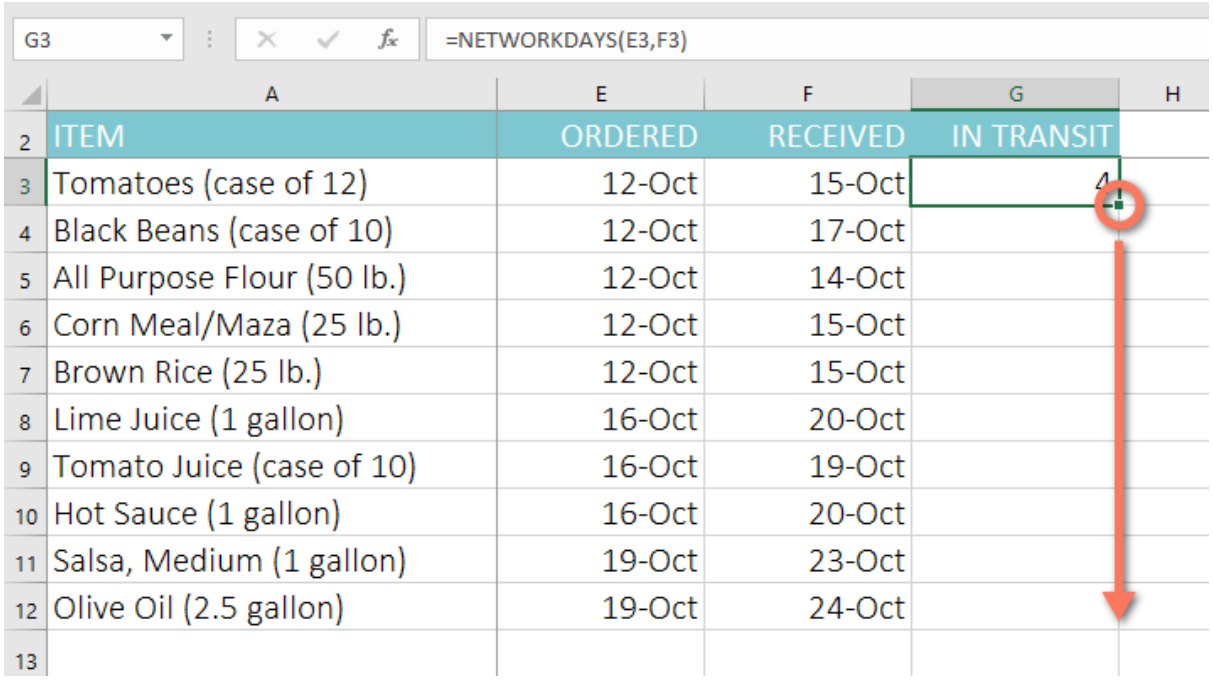



8. The function will be **calculated**, and the **result** will appear in the cell. In our example, the result shows that it took **four business days** to receive the order.




| | A | E | F | G | H |
|---|---|---|---|---|---|
| 2 | ITEM | ORDERED | RECEIVED | IN TRANSIT | |
| 3 | Tomatoes (case of 12) | 12-Oct | 15-Oct | 4 | |
| 4 | Black Beans (case of 10) | 12-Oct | 17-Oct | | |
| 5 | All Purpose Flour (50 lb.) | 12-Oct | 14-Oct | | |
| 6 | Corn Meal/Maza (25 lb.) | 12-Oct | 15-Oct | | |
| 7 | Brown Rice (25 lb.) | 12-Oct | 15-Oct | | |
| 8 | Lime Juice (1 gallon) | 16-Oct | 20-Oct | | |
| 9 | Tomato Juice (case of 10) | 16-Oct | 19-Oct | | |
| 10 | Hot Sauce (1 gallon) | 16-Oct | 20-Oct | | |
| 11 | Salsa, Medium (1 gallon) | 19-Oct | 23-Oct | | |
| 12 | Olive Oil (2.5 gallon) | 19-Oct | 24-Oct | | |
| 13 | | | | | |

Like formulas, functions can be copied to adjacent cells. Simply select the **cell** that contains the function, then click and drag the **fill handle** over the cells you want to fill. The function will be copied, and values for those cells will be calculated relative to their rows or columns.




| | A | E | F | G | H |
|---|---|---|---|---|---|
| 2 | ITEM | ORDERED | RECEIVED | IN TRANSIT | |
| 3 | Tomatoes (case of 12) | 12-Oct | 15-Oct | 4 | |
| 4 | Black Beans (case of 10) | 12-Oct | 17-Oct | | |
| 5 | All Purpose Flour (50 lb.) | 12-Oct | 14-Oct | | |
| 6 | Corn Meal/Maza (25 lb.) | 12-Oct | 15-Oct | | |
| 7 | Brown Rice (25 lb.) | 12-Oct | 15-Oct | | |
| 8 | Lime Juice (1 gallon) | 16-Oct | 20-Oct | | |
| 9 | Tomato Juice (case of 10) | 16-Oct | 19-Oct | | |
| 10 | Hot Sauce (1 gallon) | 16-Oct | 20-Oct | | |
| 11 | Salsa, Medium (1 gallon) | 19-Oct | 23-Oct | | |
| 12 | Olive Oil (2.5 gallon) | 19-Oct | 24-Oct | | |
| 13 | | | | | |

## 3.6. A Comprehensive Treatise on Excel Functionality

Excel functions constitute pre-defined formulae designed to execute complex calculations, facilitate rigorous data analysis, and automate iterative operational processes. Mastery of these instruments is requisite for the construction of robust financial models, the management of extensive datasets, and the derivation of actionable intelligence from raw information. This document elucidates the fundamental structural paradigms and categorizes the critical functions required for operational efficacy and data integrity.

### 3.6.1. 1. The Syntactical Framework

Each function adheres to a rigid, immutable structural protocol known as **syntax**. Deviation from this protocol, whether through omission of required delimiters or misplacement of arguments, results in the generation of an error state by the application, thereby halting calculation chains.

**Standard Protocol:** =FUNCTION_NAME(argument1, argument2, ...)

- **The Equality Operator (=)**: It is mandatory that every formula commences with an equality sign. This symbol signals to the application that the subsequent character string constitutes a command for processing rather than static text.
- **Function Designation (FUNCTION_NAME)**: The specific, case-insensitive nomenclature of the command (e.g., SUM, VLOOKUP, INDEX).
- **Parenthetical Enclosure (())**: Parentheses are utilized to strictly enclose the arguments. Even in instances where no arguments are required (as seen in volatile functions like TODAY), opening and closing parentheses remain compulsory.
- **Arguments**: The requisite inputs essential for the function's operation. These may take the form of direct numerical values, text strings enclosed in quotation marks, cell references (e.g., A1), or disparate ranges (e.g., A1:B10). Arguments are separated by specific delimiters—typically commas or semicolons, contingent upon regional system settings.

### 3.6.2. 1.1. Reference Modality: Absolute vs. Relative

Crucial to the deployment of functions is the distinction between relative references (e.g., A1), which adjust spatially when the formula is propagated to adjacent cells, and absolute references (e.g., $A$1), which remain fixed upon a specific coordinate regardless of replication.

### 3.6.3. 2. Mathematical and Statistical Computational Protocols

These functions serve as the foundational elements for the majority of financial valuations, statistical samplings, and quantitative analysis models.

- **SUM**

Facilitates the aggregation of all numerical values within a designated range of cells. This function is capable of processing individual cell references, contiguous ranges, or non-contiguous selections as distinct arguments.

- **Syntax:** =SUM(number1, [number2], ...)
- **Example:** =SUM(A1:A10, C1:C10) aggregates all values situated in the ranges A1 through A10 and C1 through C10 simultaneously.

- **AVERAGE**

Derives the arithmetic mean of a specified group of numerical values, calculated by dividing the aggregate sum by the count of the values.

- **Syntax:** =AVERAGE(number1, [number2], ...)
- **Example:** =AVERAGE(B2:B20) determines the central tendency of the sales figures contained within the specified column.

- **COUNT / COUNTA / COUNTBLANK**

Data quantification is achieved through three distinct variants:

- **COUNT:** Enumerates cells containing *numerical* data exclusively, disregarding text and logical values.
- **COUNTA:** Enumerates cells containing *any* data type, effectively quantifying non-null cells.
- **COUNTBLANK:** Enumerates cells that are devoid of content or contain null strings.
- **Example:** =COUNTA(A:A) quantifies the total number of populated rows within column A, providing a metric of dataset completeness.

- **MAX / MIN**

Identifies the extremes within a dataset.

- **MAX:** Returns the largest numerical value within a provided array.
- **MIN:** Returns the smallest numerical value.
- **Example:** =MAX(D2:D100) isolates the highest recorded value in the specified range.

- **ROUND**

Adjusts a numerical value to a specified degree of precision using standard rounding rules. This is critical for financial reporting where fractional cents are impermissible.

- **Syntax:** =ROUND(number, num_digits)
- **Example:** =ROUND(3.14159, 2) results in the value **3.14**.
- *Note: This differs from formatting a cell, as it permanently alters the stored value, affecting subsequent calculations.*

## 3.6.4. 3. Logical Operator Functions

These functions enable the application to execute algorithmic decision-making based upon established criteria, introducing branching logic into the spreadsheet.

- **IF**

Evaluates a specified logical condition to determine its veracity. Based upon this binary evaluation—yielding a result of TRUE or FALSE—the function returns one of two distinctive values defined by the user.

- **Syntax:** =IF(logical_test, value_if_true, value_if_false)
- **Example:** =IF(C2>10000, "Target Met", "Deficit")
    - Should the value in C2 exceed 10,000, the cell shall display "Target Met"; otherwise, the string "Deficit" shall be rendered.

- **AND / OR**

These auxiliary functions are frequently nested within the IF function to evaluate multiple conditions simultaneously.

- **AND:** Returns TRUE only if *all* arguments evaluate to TRUE.
- **OR:** Returns TRUE if *any* single argument evaluates to TRUE.
- **Example:** =IF(AND(A2>0, B2<5), "Valid", "Invalid") necessitates that both conditions be met for the "Valid" state to be returned.

- **SUMIF / COUNTIF**

Aggregates or enumerates cells contingent upon the satisfaction of specific, user-defined criteria. These functions are indispensable for conditional reporting.

- **Syntax:** =COUNTIF(range, criteria)
- **Example:** =COUNTIF(D2:D100, "Red") enumerates the quantity of cells strictly matching the string "Red".
- **Example:** =SUMIF(C2:C100, ">1000", B2:B100) aggregates values in column B *exclusively* where the corresponding value in column C exceeds 1000.

- **IFERROR**

Identifies error states arising from formula execution and substitutes them with a custom message or value. This serves as a sanitization mechanism, preventing the display of standard system error codes such as #DIV/0! or #N/A in final reports.

- **Syntax:** =IFERROR(value, value_if_error)
- **Example:** =IFERROR(A1/B1, 0) returns 0 in the event that B1 is empty or zero, thereby preserving the aesthetic and functional integrity of the sheet.

## 3.6.5. 4. Data Retrieval and Reference Mechanisms

These functions operate as a search engine for internal data, establishing relational connections between distinct tables or datasets.

- **VLOOKUP (Vertical Lookup)**

Executes a search for a specific key value in the initial (leftmost) column of a table array and returns a value in the same row from a specified column index to the right.

- **Syntax:** =VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])
- **Example:** =VLOOKUP(H2, A2:C100, 3, FALSE) locates the identifier in H2 within the range A2:C100 and retrieves the corresponding value from the 3rd column.
- *Limitation: VLOOKUP is strictly unidirectional, searching only from left to right.*

- **XLOOKUP (Advanced Retrieval Protocol)**

A contemporary, more flexible iteration of the lookup paradigm. It possesses the capability to search in any direction (vertical or horizontal), defaults to exact matches, and eliminates the risk of column index errors.

- **Syntax:** =XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])
- **Example:** =XLOOKUP(H2, A:A, C:C, "Not Found") locates the value in H2 within column A and retrieves the corresponding result from column C. If the value is absent, it returns "Not Found".

### 3.6.6. 5. String Manipulation and Sanitization Utilities

Utilized for the cleaning, normalization, and manipulation of alphanumeric text strings imported from external systems.

- **LEFT / RIGHT / MID**

Extracts a specified quantity of characters from a text string based on position.

- **Example:** =LEFT("A-123", 1) yields **"A"**, isolating the prefix.
- **Example:** =RIGHT("007", 2) yields **"07"**, isolating the suffix.
- **Example:** =MID("Part-A-001", 6, 1) yields **"A"**, extracting characters from the center of the string.

- **LEN**

Quantifies the total number of characters within a text string, including spaces and punctuation.

- **Example:** =LEN(A1) returns an integer representing the string length, useful for validating data formats (e.g., postal codes).

- **CONCAT / TEXTJOIN**

Amalgamates text from multiple cells into a single entity.

- **Syntax (CONCAT):** =CONCAT(text1, text2) (Fundamental concatenation).
- **Syntax (TEXTJOIN):** =TEXTJOIN(delimiter, ignore_empty, text1, ...)
- **Example:** =TEXTJOIN(", ", TRUE, A1:A5) generates a comma-separated list of values derived from A1 to A5, ignoring any empty cells to prevent consecutive delimiters.

- **TRIM / CLEAN**

- **TRIM:** Eliminates extraneous whitespace from text (leading, trailing, and repeated internal spaces), retaining only single spaces between words. This is optimal for the normalization of database exports.
- **CLEAN:** Removes non-printable characters often imported from legacy systems.
- **Example:** =TRIM(" Hello World ") is converted to the standardized **"Hello World"**.

### 3.6.7. 6. Temporal Calculation Functions

Essential for project management, financial amortization schedules, and the maintenance of timelines. Excel stores dates as sequential serial numbers, allowing for arithmetic operations.

- **TODAY / NOW**

Volatile functions that update upon system recalculation.

- =TODAY() retrieves the current date integer.
- =NOW() retrieves the current date and precise time fractional.

- **DATEDIF**

Calculates the differential between two dates expressed in defined units (years, months, or days).

- **Syntax:** =DATEDIF(start_date, end_date, unit)
- **Example:** =DATEDIF(A1, B1, "Y") calculates the number of complete years elapsed between the date in A1 and B1.

- **NETWORKDAYS**

Calculates the number of whole working days between two dates, automatically excluding weekends and optionally excluding a list of custom holidays.

- **Syntax:** =NETWORKDAYS(start_date, end_date, [holidays])
- **Example:** =NETWORKDAYS(A1, B1, H1:H10) determines the billable days in a period, accounting for holidays listed in range H1:H10.

## 3.6.8. 7. Diagnostic Protocols for Common Error States

Detailed analysis of error codes is required for effective troubleshooting.

| Error Code | Significance | Remediation Strategy |
|---|---|---|
| **#DIV/0!** | Division by zero | Verify the divisor cell; arithmetic principles dictate it must not be 0 or empty. Utilization of IFERROR is recommended to suppress this. |
| **#N/A** | Value unavailable | Frequently observed in Lookups when the search key is nonexistent in the source array. |
| **#NAME?** | Invalid nomenclature | It is probable that the function name has been misspelled (e.g., =SUMM(A1)) or a named range is undefined. |
| **#VALUE!** | Incorrect argument type | An attempt may be being made to perform arithmetic operations on non-numerical text cells. |
| **#REF!** | Invalid cell reference | A cell that constituted a critical part of a formula's reference chain has been deleted, breaking the link. |
| **#SPILL!** | Spill Range Blocked | A dynamic array function (like XLOOKUP) attempts to return multiple results but adjacent cells are obstructed by existing data. |

## 3.6.9. Advanced Technique: Function Nesting

It is permissible, and often necessary, to embed one function within the argument of another. This practice is referred to as **nesting**. Excel evaluates these formulae from the innermost parentheses outward.

- **Example:** =IF(SUM(A1:A10)>100, "Goal Met", "Keep Trying")
    - **Step 1:** The internal SUM(A1:A10) function is executed first to derive a total.
    - **Step 2:** The IF function subsequently evaluates whether said total exceeds the threshold of 100.
    - **Step 3:** The final text string is returned based on the logical outcome.