

# EXCEL TRAINING

## Table of Contents

<b>Module 1 – Introduction .....</b>	<b>4</b>
What is Microsoft Excel? .....	4
Microsoft Excel Interface.....	5
Formula tab and Auditing tools .....	5
Formula Syntax.....	6
Example .....	6
Reference Type .....	6
Example (Create a multiplication table).....	7
Common Errors.....	8
IF ERROR.....	8
Example (Division by 0). 0/0 .....	9
Shortcuts .....	9
Using the CTRL FUNCTIONS .....	10
ALT KEYS.....	10
DATA VALIDATION .....	10
Example (Create Months).....	11
<b>Module 2 – Logical Operators .....</b>	<b>12</b>
If Statement.....	13
EXAMPLE.....	14
NESTED IF .....	15
NOT / <>.....	16
IF ERROR.....	17
IS ERROR.....	18
<b>Module 3 – Statistical Function .....</b>	<b>20</b>
Commonly Used Statistical Functions .....	21
EXAMPLE.....	21
RANK.....	25
PERCENTRANK .....	26
EXAMPLE.....	26
RAND() AND RANDBETWEEN() .....	27
EXAMPLE.....	27
Sum product .....	28
EXAMPLE.....	30
<b>Module 4 – Lookup / Reference Functions .....</b>	<b>32</b>
Named Arrays.....	33

EXAMPLE.....	33
<b>ROW/ROWS .....</b>	<b>35</b>
<b>Column / Columns .....</b>	<b>36</b>
EXAMPLE.....	36
<b>VLOOKUP.....</b>	<b>38</b>
<b>HLOOKUP .....</b>	<b>38</b>
<b>LAWS OF LOOKUPS .....</b>	<b>39</b>
EXAMPLE.....	39
<b>INDEX.....</b>	<b>42</b>
<b>MATCH.....</b>	<b>42</b>
<b>INDEX / MATCH .....</b>	<b>42</b>
EXAMPLE.....	43

## Module 1 – Introduction

Objectives.

At the end of the module the participant is expected to:

1. Define Microsoft Excel.
2. Identify the different interface of the Microsoft Excel application.
3. Navigate the Library function and audit tools
4. Apply the Fixed and relative reference
5. List and apply the common errors in excel
6. Define data validation and apply it.

Topics:

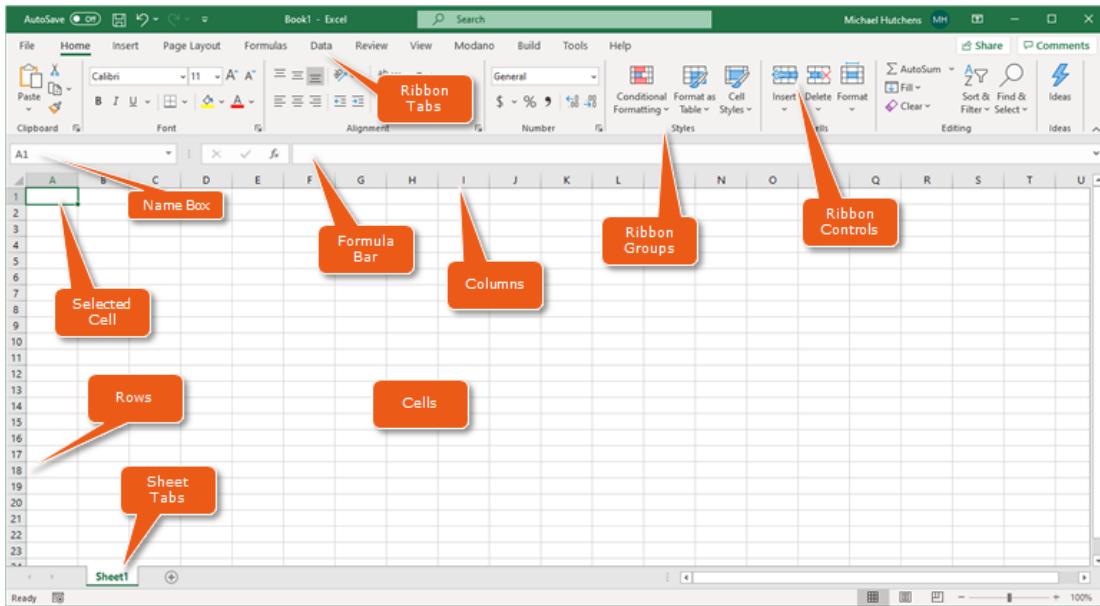
1. Library Functions and Audit Tools
2. Fixed and Relative References
3. Common errors
4. Data Validation

### What is Microsoft Excel?

Microsoft Excel is a spreadsheet powerful program for data analysis and documentation.

It columns and rows, where each intersection of a column and a row is a “cell.” Each cell contains one point of data or one piece of information. By organizing the information in this way, you can make information easier to find, and automatically draw information from changing data.

## Microsoft Excel Interface



## Formula tab and Auditing tools

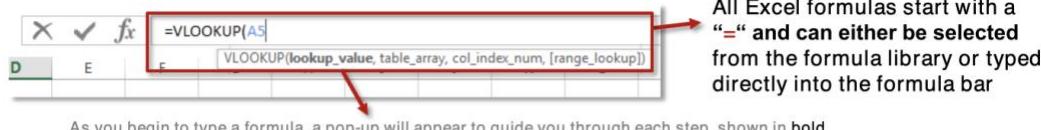
The screenshot shows the Microsoft Excel ribbon with the **Formulas** tab selected. A red box highlights the **Defined Names** and **Formula Auditing** sections. An arrow points from the **FORMULA LIBRARY** section below to the **Logical** function category in the ribbon.

**FORMULA LIBRARY:**  
Includes a list of all common formulas, component parts, and brief descriptions of how each formula works

**AUDITING TOOLS:**

- Trace Precedents/Dependents** shows which cells affect or are affected by the value of the selected cell
- Show Formulas** displays all of the formulas in the sheet as text
- Evaluate Formula** allows you to step into a formula and determine the output of each component

## Formula Syntax



As you begin to type a formula, a pop-up will appear to guide you through each step, shown in bold

**A1**

**Single-cell references** describe a cell's location within a worksheet, in terms of the intersection between a column (A through XFD), and a row (1 through 1,048,576)

A	B	C
1	2	3
2	3	4
3	4	5
4	5	6
5	6	7
6	7	8
7	8	9
8	9	10
9	10	
10		

**A1:C4**

**Array references** describe a contiguous group of cells based on the location of the top-left (A1) and the bottom-right (C4) cells, separated by a “:”

A	B	C
1	2	3
2	3	4
3	4	5
4	5	6
5	6	7
6	7	8
7	8	9
8	9	10
9	10	
10		

**A1,C4**

**Non-contiguous references** describe selections of individual cells that do not share a common border, separated by a “,”

A	B	C
1	2	3
2	4	
3		
4		

## Example

Single cell reference	Array Reference	Non Contiguous reference PRESS CONTROL or not

## Reference Type

### Hold the phone, how come some cell references include a “\$”?

These are used to create **Fixed, Relative, or Mixed References**; the \$ basically locks a specific cell range or reference so that it does not change if you apply the formula to other cells.

For Example:

- \$A\$1 = Fixed column, Fixed row
- A\$1 = Relative column, Fixed row
- \$A1 = Fixed column, Relative row
- A1 = Relative column, Relative row

A	B	C
1	\$A\$1	
2		
3		
4		

A	B	C
1	A\$1	
2		
3		
4		

#### PRO TIP:

Select part of your formula with the cursor and use “F4” to quickly scroll through reference types. ALWAYS THINK ABOUT YOUR REFERENCES

A	B	C
1	\$A1	
2		
3		
4		

A	B	C
1	A1	
2		
3		
4		

### Example (Create a multiplication table)

- Fill contents with 1-10 for rows and column, multiply it using the formula =b1\*a1

	A	B	C	D	E	F	G	H	I	J	K
1		1	2	3	4	5	6	7	8	9	10
2	1	1									
3	2		2								
4	3			4							
5	4				5						
6	5					6					
7	6						7				
8	7							8			
9	8								9		
10	9									10	
11	10										

- Populate the values, by copying the formula

	A	B	C	D	E	F	G	H	I	J	K	
1			1	2	3	4	5	6	7	8	9	10
2	1	1	2	6	24	120	720	5040	40320	362880	3628800	
3	2	2	4	24	576	69120	49766400	2.5082E+11	1.0113E+16	3.6699E+21	1.3317E+28	
4	3	6	24	576	331776	2.2932E+10	1.1413E+18	2.8625E+29	2.8949E+45	1.0624E+67	1.4148E+95	
5	4	24	576	331776	1.1008E+11	2.5243E+21	2.8809E+39	8.2466E+68	2.387E+114	2.536E+181	3.588E+276	
6	5	120	69120	2.2932E+10	2.5243E+21	6.372E+42	1.8357E+82	1.514E+151	3.614E+265	#NUM!	#NUM!	
7	6	720	49766400	1.1413E+18	2.8809E+39	1.8357E+82	3.37E+164	#NUM!	#NUM!	#NUM!	#NUM!	
8	7	5040	2.5082E+11	2.8625E+29	8.2466E+68	1.514E+151	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	
9	8	40320	1.0113E+16	2.8949E+45	2.387E+114	3.614E+265	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	
10	9	362880	3.6699E+21	1.0624E+67	2.536E+181	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	
11	10	3628800	1.3317E+28	1.4148E+95	3.588E+276	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	#NUM!	

- It will produce incorrect computation of multiplication table
- Select any cell check the data, it's not computing the correct reference value.

C	D	E
2	3	4
2	6	24
4	24	576
24	576	=E3*D4
576	331776	1.1008E+11

- To correct the computation use the formula (**=B\$1\*\$A2**). Make the row and column absolute.

B2	A	B	C	D	E	F	G	H	I	J	K
	1	1	2	3	4	5	6	7	8	9	10
1	1	1	2	3	4	5	6	7	8	9	10
2	2	2	4	6	8	10	12	14	16	18	20
3	3	3	6	9	12	15	18	21	24	27	30
4	4	4	8	12	16	20	24	28	32	36	40
5	5	5	10	15	20	25	30	35	40	45	50
6	6	6	12	18	24	30	36	42	48	54	60
7	7	7	14	21	28	35	42	49	56	63	70
8	8	8	16	24	32	40	48	56	64	72	80
9	9	9	18	27	36	45	54	63	72	81	90
10	10	10	20	30	40	50	60	70	80	90	100
11											

## Common Errors

Error Type	What it means	How to fix it
#####	Column isn't wide enough to display values	Drag or double-click column border to increase width, or right-click to set custom column width
#NAME?	Excel does not recognize text in a formula	Make sure that function names are correct, references are valid and spelled properly, and quotation marks and colons are in place
#VALUE!	Formula has the wrong type of argument	Check that your formula isn't trying to perform an arithmetic operation on text strings or cells formatted as text
#DIV/0!	Formula is dividing by zero or an empty cell	Check the value of your divisor; if 0 is correct, use an IF statement to display an alternate value if you choose
#REF!	Formula refers to a cell that is not valid	Make sure that you didn't move, delete, or replace cells that are referenced in your formula

## IF ERROR

The **IFERROR** statement is an excellent tool to eliminate annoying error messages (#N/A, #DIV/0!, #REF!, etc.), which is particularly useful for front-end formatting

=IFERROR(value, value\_if\_error)

Formula or value (which may or may not result in an error)

Value returned in the case of an error

In this case we're replacing an error caused by the A1/B1 formula with "Invalid Formula", and an error caused by a VLOOKUP function with "-"

=IFERROR(A1/B1,"Invalid Formula")  
=IFERROR(VLOOKUP(A1,D1:E4,2,0),"")



### PRO TIP:

If you're writing a formula that may trigger an error (i.e. a VLOOKUP where not all values have a match), WRITE THE FULL FORMULA FIRST then wrap it in an IFERROR statement

Example (Division by 0). 0/0

L7	G	H	I	J	K	L	M	N
3								
4								
5								
6								
7	0	0	0	0	-	#DIV/0!		
8								

## Shortcuts

The **F4** function is used for two helpful shortcuts:

- 1) Adding or modifying cell reference types

With your cursor selecting any cell reference or array within a formula, the F4 key will cycle through fixed, relative, and absolute reference types



- 2) Repeating your last command or action ↵

F4 will also repeat the last user action, such as inserting/deleting rows or columns, changing cell format or style, etc. (Note: F4 will not repeat entered values or formulas)

The **F2** function displays the cell ranges that are tied to a given formula



### PRO TIP:

Use **F2** to help diagnose formula errors or make quick adjustments to cell references and arrays

## Using the CTRL FUNCTIONS

The **CTRL** function can be combined with a variety of keys, such as:

### 1) CTRL- ARROW

Jumps to the left, right, top, or bottom edge (i.e. last non-blank cell) of a contiguous data array

A	B	C	D	E	F	G	H
1	77	847	482	847	916	329	796
2	181	852	286	275	177	476	234
3	232	137	234	343	342	343	343
4	711	507	125	930	348	529	491
5	842	12	837	493	221	398	398
6	302	89	390	245	295	751	944
7	826	676	479	777	171	441	104
8	123	264	893	893	893	893	893
9	376	447	395	232	18	77	847
10	851	966	861	898	75	183	852
11	738	775	651	817	9	213	137
12					4	711	507

### 2) CTRL-SHIFT-ARROW

Extends a selection to the left, right, top, or bottom edge (i.e. last non-blank cell) of a data array

### 3) CTRL-PAGE UP/DOWN

Jumps between tabs of a workbook

A	B	C	D	E	F	G	H
1	77	847	482	847	916	329	796
2	181	852	286	275	177	476	234
3	232	137	234	343	342	343	343
4	711	507	125	930	348	529	491
5	842	12	837	493	221	398	398
6	302	89	390	245	295	751	944
7	826	676	479	777	171	441	104
8	123	264	893	893	893	893	893
9	376	447	395	232	172	655	984
10	851	966	861	898	225	560	652
11	738	775	651	817	550	703	662
12					8	803	461

## ALT KEYS

The **ALT** function enables **Key Tips**, which allow you to access any function in the ribbon using keyboard shortcuts (Note: you do not need to hold down ALT)

- 1) Press **ALT** to reveal tab-level shortcuts:



- 2) Press the key for the tab you want (i.e. **H**) to reveal additional shortcuts:



- 3) Continue to press shortcut keys (i.e. **V**) to drill into specific functions:



## DATA VALIDATION

**Data Validation** allows you to specify exactly what types of values a cell can contain (i.e. whole numbers, positive integers, values from a list, etc.)

One of the most useful forms of data validation is **LIST**, which creates a drop-down menu of options based on a source list that you specify:

(but the best part is that you can write your own hilarious error messages) See, Excel can be fun!



**Data Validation**

**Settings**   **Input Message**   **Error Alert**

**Validation criteria**

**Allow:** **List**  **Ignore blank**  **In-cell dropdown**

**Data:** **between**

**Source:** **=SL\$4:SL\$9**

**Apply these changes to all other cells with the same settings**

**OK**   **Cancel**

**PRO TIP:** Use **ALT-H-V-V** to paste as values or **ALT-H-V-F** to paste as formulas

**J**   **K**   **L**

C
A
B
C
D
E
F

**List:**

A
B
C
D
E
F

### Example (Create Months)

- Select (DATA) from the ribbon, then Data Validation.

Months
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

## Module 2 – Logical Operators

Objectives.

At the end of the module the participant is expected to:

1. Understand the logical operator in MS Excel.
2. Apply the following
  - a. IF statement
  - b. Nested if
  - c. AND / OR Operators
  - d. Not Operator
  - e. IF Error
  - f. IS Statements

Topics:

1. Anatomy of the IF statement
2. Nested Logical Operators
3. And / Or Operator
4. Not VS <> Operator
5. IS statement

A logical operator is used in Excel to compare two values. Logical operators are sometimes called Boolean operators because the result of the comparison in any given case can only be either TRUE or FALSE.



Condition	Operator	Formula Example	Description
Equal to	=	=A1=B1	The formula returns TRUE if a value in cell A1 is equal to the values in cell B1; FALSE otherwise.
Not equal to	$\diamond$	=A1 $\diamond$ B1	The formula returns TRUE if a value in cell A1 is not equal to the value in cell B1; FALSE otherwise.
Greater than	>	=A1>B1	The formula returns TRUE if a value in cell A1 is greater than a value in cell B1; otherwise it returns FALSE.
Less than	<	=A1<B1	The formula returns TRUE if a value in cell A1 is less than in cell B1; FALSE otherwise.
Greater than or equal to	$\geq$	=A1 $\geq$ B1	The formula returns TRUE if a value in cell A1 is greater than or equal to the values in cell B1; FALSE otherwise.
Less than or equal to	$\leq$	=A1 $\leq$ B1	The formula returns TRUE if a value in cell A1 is less than or equal to the values in cell B1; FALSE otherwise.

## If Statement

The IF function is one of the most popular functions in Excel, and it allows you to make logical comparisons between a value and what you expect.

So an IF statement can have two results. The first result is if your comparison is True, the second if your comparison is False.

$=IF(logical\_test, [Value\ if\ True], [Value\ if\ False])$

Any test that results in either  
TRUE or FALSE  
(i.e.  $A1="Google"$ ,  $B2<100$ , etc)

Value returned if logical  
test is TRUE

Value returned if logical  
test is FALSE

	A	B	C	D
1	Location	Temp (F)	Precip (mm)	Freeze
2	A	75	0	No
3	B	18	0	Yes
4	C	86	0	No
5	D	80	2.3	No
6	E	28	1.2	Yes
7	F	68	0.5	No
8	G	26	0	Yes

$= IF(B2 \leq 0, "Yes", "No")$

In this case we're categorizing the Freeze column as "Yes" if the temperature is equal to or below 32, otherwise "No"

## EXAMPLE

- Open the DEMO\_MODULE2.XLSX File

J8	B	C	D	E	F	G	H	I	J	K	L	M	N
	STATION_NAME	LAT	LONG	DATE	MONTH	TEMP (F)	PRCP (mm)	Freeze	Climate	PRCP Type	Conditions	Missing Data?	
1	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100101	01	37	0						
2	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100102	01	12	109						
3	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100104	01	34	0						
4	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100105	01								
5	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100106	01	20	0						
6	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100107	01	10	0						
7	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100108	01	31	3						
8	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100109	01	17	0						
9	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100110	01	12	0						
10	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100111	01	#N/A	0						
11	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100112	01	28	0						
12	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100113	01	43	0						
13	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100114	01								
14	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100115	01	43	0						
15	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100116	01	38	0						
16	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100117	01	11	109						
17	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100118	01	12	180						
18	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100119	01	37	76						
19	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100120	01	27	0						
20	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100121	01	41	0						
21	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100122	01	19	0						
22	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100123	01	39	0						
23	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100124	01	25	0						
24	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100125	01	40	206						
25	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100126	01	37	0						
26	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100127	01	18	0						
27	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100128	01	35	5						
28	BOSTON LOGAN AIRPORT	42.3606	-71.0106	20100129	01	30	0						

- Start with Column I (FREEZE COLUMN)

Condition if TEMP (COLUMN G) is less than 32 (YES) Otherwise (NO)

```
=IF(G2<32,"Yes","No")
IF(logical_test, [value_if_true], [value_if_false])
```

Populate the remaining values down.

NOTE: Don't mind with missing data.

## NESTED IF

By using **Nested IF Statements**, you can include multiple logical tests within a single formula:

	A	B	C	D	E
1	Location	Temp (F)	Precip (mm)	Freeze	Climate
2	A	75	0	No	Mild
3	B	18	0	Yes	Cold
4	C	86	0	No	Hot
5	D	80	2.3	No	Mild
6	E	28	1.2	Yes	Cold
7	F	68	0.5	No	Mild
8	G	26	0	Yes	Cold

= IF(B2<40,"COLD",IF(B2>80,"HOT","MILD"))  
*If temp<40, climate = "Cold", if temp>80, climate = "Hot", otherwise climate = "Mild"*

- REFER TO COLUMN J (CLIMATE)

G	H	I	J	K
TEMP (F)	PRCP (mm)	Freeze	Climate	PRCP Type
37			=IF(G2<40,"Cold",IF(G2>80,"Hot","MILD"))	
12	109	Yes		+

- Populate the remaining values down.

Excel's **AND** and **OR** statements allow you to include multiple logical tests at once:

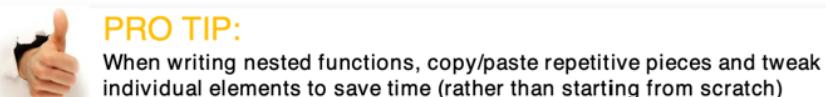
A	B	C	D	E	F	G	
1	Location	Temp (F)	Precip (mm)	Freeze	Climate	Precip Type	Conditions
2	A	75	0	No	Mild	None	Dry
3	B	18	0	Yes	Cold	None	Dry
4	C	86	0	No	Hot	None	Dry
5	D	80	2.3	No	Mild	Rain	Wet
6	E	28	1.2	Yes	Cold	Snow	Wet
7	F	68	0.5	No	Mild	Rain	Wet
8	G	26	0	Yes	Cold	None	Dry

=IF(OR(F2="Rain",F2="Snow"),"Wet","Dry")  
*Here we're categorizing conditions as "Wet" if the precipitation type equals "rain" OR "snow", otherwise Conditions = "Dry"*

=IF(AND(D2="Yes",C2>0),"Snow",IF(AND(D2="No",C2>0),"Rain","None"))

If the temp is below freezing AND the amount of precipitation > 0, then Precip Type = "Snow", if the temp is above freezing AND the amount of precipitation >0, then Precip Type = "Rain", otherwise Precip Type = "None"

### PRO TIP:



- REFER TO THE PRCP Type column

H	I	J	K	L	M	N
PRCP (mm)	Freeze	Climate	PRCP Type	Conditions	Missing Data?	
			=IF(AND(I2="Yes",H2>0),"Snow",IF(AND(I2="No",H2>0),"Rain","None"))			
109			IF(logical_test, [value_if_true], [value_if_false])			

- Refer to (CONDITION COLUMN)

K	L	M	N
PRCP Type	Conditions	Missing Data?	
	=IF(OR(K2="Snow",K2="Rain"),"Wet","Dry")		
	IF(logical_test, [value_if_true], [value_if_false])		
	Snow		

NOT / <>

If you want to evaluate a case where a logical statement is not true, you can use either the NOT statement or a “ $\neq$ ” operator

A	B	C	D	E	F	G
Location	Temp (F)	Precip (mm)	Freeze	Climate	Precip Type	Conditions
A	75	0	No	Mild	None	Dry
B	18	0	Yes	Cold	None	Dry
C	86	0	No	Hot	None	Dry
D	80	2.3	No	Mild	Rain	Wet
E	28	1.2	Yes	Cold	Snow	Wet
F	68	0.5	No	Mild	Rain	Wet
G	26	0	Yes	Cold	None	Dry

=IF(NOT(C2=0), "Wet", "Dry")  
=IF(C2<>0, "Wet", "Dry")

In both of these examples, we're defining Conditions = "Wet" if the amount of precipitation is NOT equal to 0

- REFER TO COLUMN L
- Condition COLUMN

H	I	J	K	L	M	
PRCP (mm)	Freeze	Climate	PRCP Type	Conditions	Missing	
0	No	Cold		=IF(H2<>0, "Wet", "Dry")		
109	Yes	Cold	Snow		Wet	
51	Yes	Cold	Snow		Wet	

## IF ERROR

The **IFERROR** statement is an excellent tool to eliminate annoying error messages (#N/A, #DIV/0!, #REF!, etc.), which is particularly useful for front-end formatting

=IFERROR(value, value\_if\_error)

Formula or value that may or may not result in an error

Value returned in the case of an error



### PRO TIP:

If you're writing a formula that may trigger an error (i.e. a VLOOKUP where not all values have a match), WRITE THE FULL FORMULA FIRST then wrap it in an IFERROR statement

- REFER TO COLUMN I (FREEZE COLUMN)
- Update the formula

I	Freeze	C1
	No	(
	Yes	(
	Yes	)

=IFERROR(IF(G2<32,"Yes","No"),"Other")

- REFER TO COLUMN J (CLIMATE COLUMN)
- Update the formula

C	IFERROR(value, value_if_error)		G	H	I	J	
LAT	LONG	DATE	MONTH	TEMP (F)	PRCP (mm)	Freeze	Climate
2.3606	-71.0106	20100101	01	37	0	No	"")),"Other")
2.3606	-71.0106	20100102	01	12	109	Yes	Cold
2.3606	-71.0106	20100103	01	23	51	Yes	Cold

## IS ERROR

Excel offers a number of different **IS** formulas, each of which checks whether a certain condition is true:

**ISBLANK** = Checks whether the reference cell or value is blank

**ISNUMBER** = Checks whether the reference cell or value is numerical

**ISTEXT** = Checks whether the reference cell or value is a text string

**ISERROR** = Checks whether the reference cell or value returns an error

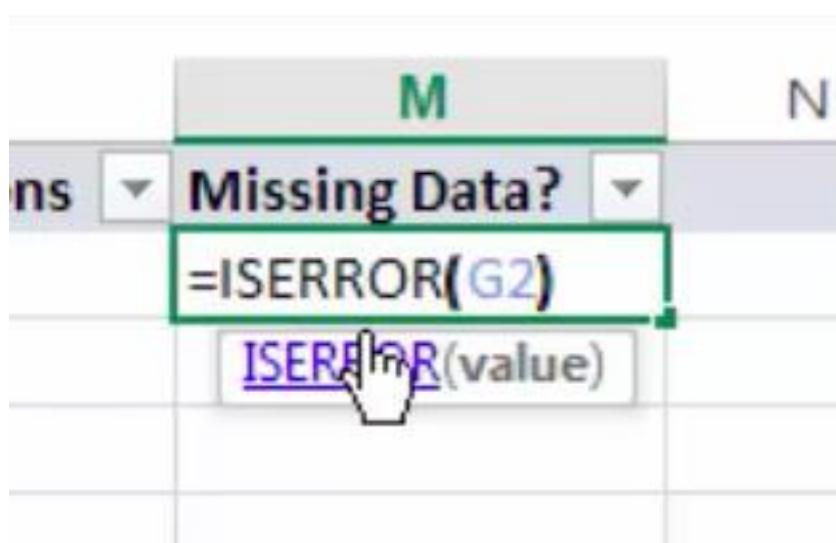
**ISEVEN** = Checks whether the reference cell or value is even

**ISODD** = Checks whether the reference cell or value is odd

**ISLOGICAL** = Checks whether the reference cell or value is a logical operator

**ISFORMULA** = Checks whether the reference cell or value is a formula

- REFER TO COLUMN M (MISSING DATA)



<b>M</b> <b>Missing Data?</b> <b>=ISBLANK(G2)</b> <small>ISBLANK(value)</small> <b>FALSE</b>
--

- Combine formula

<b>Missing Data?</b> <b>=OR(ISBLANK(G2), ISERROR(G2))</b> <small>+ logical1, [logical2], [logical3], ...</small> <b>FALSE</b>
--

## Module 3 – Statistical Function

Objectives.

At the end of the module the participant is expected to:

1. Understand and apply different statistical function in excel
  - a. Count, Average, Median, Max/Min, Percentile, Standard Deviation/Variance
  - b. Rank / Small
  - c. Percent Rank
  - d. Rand /Rand Between
  - e. Sum Products
  - f. Count ifs / Avarage Ifs

Topics:

1. Basic statistics functions
2. Small/Large, Rank/Percentrank
3. Rand(), RandBetween()
4. Sum Products
5. Countif Sumif / Average If

In addition to formulas, another way to conduct mathematical computations in Excel is through functions. Statistical functions apply a mathematical process to a group of cells in a worksheet.

## Commonly Used Statistical Functions

Function	Output
<b>ABS</b>	The absolute value of a number
<b>AVERAGE</b>	The average or arithmetic mean for a group of numbers
<b>COUNT</b>	The number of cell locations in a range that contain a numeric character
<b>COUNTA</b>	The number of cell locations in a range that contain a text or numeric character
<b>MAX</b>	The highest numeric value in a group of numbers
<b>MEDIAN</b>	The middle number in a group of numbers (half the numbers in the group are higher than the median and half the numbers in the group are lower than the median)
<b>MIN</b>	The lowest numeric value in a group of numbers
<b>MODE</b>	The number that appears most frequently in a group of numbers
<b>PRODUCT</b>	The result of multiplying all the values in a range of cell locations
<b>SQRT</b>	The positive square root of a number
<b>STDEV.S</b>	The standard deviation for a group of numbers based on a sample
<b>SUM</b>	The total of all numeric values in a group

The Count, Average, Median, Mode, Max/Min, Percentile and Standard Deviation/Variance functions are used to perform basic calculations on a data array

	A	B	C	D	
1					
2	Value				
3	90	Sample Size	19		=COUNT(A2:A20)
4	13				
5	22	Average:	51.47		=AVERAGE(A2:A20)
6	98				
7	61	Median:	54		=MEDIAN(A2:A20)
8	68				
9	50	Mode:	22		=MODE(A2:A20)
10	91				
11	16	Max:	98		=MAX(A2:A20)
12	23				
13	60	Min:	13		=MIN(A2:A20)
14	22				
15	56	25th Percentile	23		=PERCENTILE(A2:A20, .25)
16	54				
17	87	75th Percentile	68		=PERCENTILE(A2:A20, .75)
18	33				
19	68	Standard Deviation	28		=STDEV(A2:A20)
20	45				
21	21	Variance	767		=VAR(A2:A20)

### EXAMPLE

- Open the DEMO\_MODULE3.XLSX file
- Select Basic Stats Functions (sheet)

COUNT - SAMPLE SIZE

```
=COUNT('Salary Data'!E:E)
```

D		+
ue	Player	Salary

AVERAGE

```
=AVERAGE('Salary Data'!E:E)
```

D		+
ue	Player	Salary
	abreuto01	\$407,000

MEDIAN

```
=MEDIAN('Salary Data'!E:E)
```

D	E	+
ue	Player	Salary
	abreuto01	\$407,000
	boyerbl01	\$725,000

MODE

```
=MODE('Salary Data'!E:E)
```

D	E	
ue	Player	Salary
	abreuto01	\$407,000
	boyerbl01	\$725,000

MAX

```
=MAX('Salary Data'!E:E)
```

	D	E
Player		Salary
abreuto01		\$407,000
boyerbl01		\$725,000

MIN

```
=MIN('Salary Data'!E:E)
```

	D	E
Player		Salary
abreuto01		\$407,000
boyerbl01		\$725,000

PERCENTILE .75

```
=PERCENTILE('Salary Data'!E:E,.75)
```

PERCENTILE(array, k)

	D	E
Player		Salary
abreuto01		\$407,000
boyerbl01		\$725,000

*PERCENTILE .25*

```
=PERCENTILE('Salary Data'!E:E,0.25)
```

PERCENTILE(array, k) C

D

*STDEV*

```
=STDEV('Salary Data'!E:E)
```

	D	E
ue	Player	Salary
	abreuto01	\$407,000

*VAR*

```
=VAR('Salary Data'!E:E)
```

	D	+
gue	Player	Salary
	abreuto01	\$407,000

## RANK

A	Value
1	90
2	13
3	22
4	98
5	61
6	68
7	50
8	

The **SMALL/LARGE** functions return the  $n^{\text{th}}$  smallest/largest values within an array

The **RANK** function returns the rank of a particular number among a list of values

A	Value
1	90
2	13
3	22
4	98
5	61
6	68
7	50
8	

**LARGE(A2:A8,2) = 90**

(the 2nd largest number in the array is 90)

**SMALL(A2:A8,3) = 50**

(the 3rd smallest number in the array is 50)

## LARGE

=LARGE('Salary Data'!E:E,10)

D	E
Player	Salary
abreuto01	\$407,000

## SMALL

=SMALL('Salary Data'!E:E,100)

D	E
Player	Salary
abreuto01	\$407,000

## PERCENTRANK

	A	B
1	Value	Percent Rank
2	2,717	18%
3	3,485	24%
4	5,202	76%
5	3,612	29%
6	4,432	59%
7	2,699	12%
8	4,585	65%
9	6,003	94%
10	4,820	71%
11	2,550	6%
12	5,795	88%
13	4,240	41%
14	6,827	100%
15	4,359	53%
16	2,320	0%
17	5,775	82%
18	4,241	47%
19	3,966	35%

**PERCENTRANK** returns the rank of a value as a percentage of a given array or dataset

=PERCENTRANK(array, x)

What range of data are you looking at?

Which value within the range are you looking at?

PERCENTRANK(A14,\$A\$2:\$A\$19) = 100% (highest)

PERCENTRANK(A14,\$A\$2:\$A\$19) = 0% (lowest)

## EXAMPLE

- SELECT SALARY SHEET

Year	Team	League	Player	Salary	Rank	% Rank
2010	ARI	NL	abreuto01	\$407,000		
2010	ARI	NL	boyerbl01	\$725,000		
2010	ARI	NL	drewst01	\$3,400,000		
2010	ARI	NL	gutieju01	\$411,000		
2010	ARI	NL	harenda01	\$8,250,000		
2010	ARI	NL	heilmaa01	\$2,150,000		
2010	ARI	NL	howrybo01	\$2,000,000		
2010	ARI	NL	jacksco01	\$3,100,000		
2010	ARI	NL	iacksed01	\$4,600,000		

=RANK(E2,\$E\$2:\$E\$4135)

ue	D	E	F
Player	Salary	Rank	% R
abreuto01			
boyerbl01	\$725,000	=RANK(E2,\$E\$2:\$E\$4135)	

						G		
Due	Player	Salary	Rank	% Rank			H	I
L	abreuto01	\$407,900	4,014	=PERCENTRANK(\$E\$2:\$E\$4135,E2)				
L	boyerbl01	\$725,000	2,457	PERCENTRANK(array, x, [significance])				

## RAND() AND RANDBETWEEN()

RAND() and RANDBETWEEN act like random number generators in Excel:

	A	B	C	D	E
1	0.5173	0.4091	0.7560	0.9012	0.2167
2	0.0906	0.2317	0.0906	0.5856	0.8646
3	0.1544	0.8240	0.4279	0.8782	0.7795
4	0.0097	0.0872	0.7740	0.9137	0.7815
5	0.2089	0.7028	0.0449	0.8173	0.9983
6	0.0761	0.4388	0.4056	0.5639	0.0668

The RAND() function returns a random value between 0 and 1 (to 15 digits)

The RANDBETWEEN function returns an integer between two values that you specify

	A	B	C	D	E
1	83	23	64	62	92
2	59	45	40	50	91
3	24	37	70	30	32
4	54	85	69	55	3
5	73	12	36	53	2
6	29	72	68	59	99

=RANDBETWEEN(0,100)

## EXAMPLE

- CREATE A BLANK SHEET

A1						=RAND()
1	0.448363	0.056526	0.614612	0.277134	0.580765	0.555411
2	0.005657	0.577811	0.839086	0.577575	0.201978	0.493911
3	0.815942	0.287145	0.605143	0.581204	0.378896	0.699231
4	0.203223	0.443947	0.021644	0.751117	0.571093	0.712581
5	0.108417	0.003739	0.099357	0.633722	0.201663	0.520961
6	0.144305	0.647357	0.914601	0.095292	0.46071	0.592291

## RANDBETWEEN()

A1						=RANDBETWEEN(0,100)
1	13	47	51	91	49	33
2	20	64	17	81	63	48
3	82	46	77	67	14	19
4	92	58	77	62	11	96
5	88	44	6	88	47	60
6	49	54	48	5	85	31

## Sum product

The **SUMPRODUCT** formula multiplies corresponding cells from multiple arrays and returns the sum of the products (Note: all arrays must have the same dimensions)

=SUMPRODUCT(array1, array2 ... array\_N)

Example: Total Revenue

	A	B	C	D
1	Product	Quantity	Price	Revenue
2	Apple	2	\$0.50	\$1.00
3	Banana	4	\$1.00	\$4.00
4	Orange	3	\$0.80	\$2.40
5	Total			\$7.40

	A	B	C	D
1	Product	Quantity	Price	Revenue
2	Apple	2	\$0.50	
3	Banana	4	\$1.00	
4	Orange	3	\$0.80	
5	Total			\$7.40

Without using SUMPRODUCT, you could multiply quantity\*price in each row and sum the products

SUMPRODUCT(B2:B4,C2:C4) = \$7.40

**SUMPRODUCT** is often used with filters to calculate products only for rows that meet certain criteria:

	A	B	C	D
1	Store	Product	Quantity	Price
2	Stop & Shop	Apple	2	\$0.50
3	Shaws	Banana	4	\$1.00
4	Market Basket	Banana	3	\$1.00
5	Trader Joe's	Pineapple	8	\$2.50
6	Stop & Shop	Orange	2	\$0.80
7	Shaws	Apple	1	\$0.50
8	Market Basket	Apple	5	\$0.50
9	Trader Joe's	Banana	6	\$1.00
10	Market Basket	Pineapple	3	\$2.50
11	Trader Joe's	Orange	8	\$0.80
12	Stop & Shop	Pineapple	3	\$2.50
13	Shaws	Pineapple	5	\$2.50
14	Stop & Shop	Banana	2	\$1.00
15	Shaws	Orange	6	\$0.80
16	Market Basket	Orange	7	\$0.80
17	Trader Joe's	Apple	3	\$0.50

Quantity of goods sold at Shaws:

SUMPRODUCT((A2:A17="Shaws")\*C2:C17) = 16

Total revenue from Shaws:

SUMPRODUCT((A2:A17="Shaws")\*C2:C17\*D2:D17) = \$21.80

Revenue from apples sold at Shaws:

SUMPRODUCT((A2:A17="Shaws")\*(B2:B17="Apple")\*C2:C17\*D2:D17) = \$0.50



### PRO TIP:

When you add filters to a SUMPRODUCT, you need to change the commas to multiplication signs

Great, but how does it really work?

`SUMPRODUCT((A2:A17="Shaws")*(B2:B17="Apple")*C2:C17*D2:D17) = $0.50`

A	B	C	D
Store	Product	Quantity	Price
Stop & Shop	Apple	2	\$0.50
Shaws	Banana	4	\$1.00
Market Basket	Banana	3	\$1.00
Trader Joe's	Pineapple	8	\$2.50
Stop & Shop	Orange	2	\$0.80
Shaws	Apple	1	\$0.50
Market Basket	Apple	5	\$0.50
Trader Joe's	Banana	6	\$1.00
Market Basket	Pineapple	3	\$2.50
Trader Joe's	Orange	8	\$0.80
Stop & Shop	Pineapple	3	\$2.50
Shaws	Pineapple	5	\$2.50
Stop & Shop	Banana	2	\$1.00
Shaws	Orange	6	\$0.80
Market Basket	Orange	7	\$0.80
Trader Joe's	Apple	3	\$0.50



When you apply a condition or filter to a column, Excel translates those cells as 0's (if false) and 1's (if true)

If you multiply all four columns,  
ONLY ROWS THAT SATISFY  
ALL CONDITIONS WILL  
PRODUCE A NON-ZERO SUM

A	B	C	D
Store	Product	Quantity	Price
0	1	2	\$0.50
1	0	4	\$1.00
0	0	3	\$1.00
0	0	8	\$2.50
0	0	2	\$0.80
1	1	1	\$0.50
0	1	5	\$0.50
0	0	6	\$1.00
0	0	3	\$2.50
0	0	8	\$0.80
0	0	3	\$2.50
1	0	5	\$2.50
0	0	2	\$1.00
1	0	6	\$0.80
0	0	7	\$0.80
0	1	3	\$0.50

The **COUNTIF**, **SUMIF**, and **AVERAGEIF** formulas calculate a sum, count, or average based on specific criteria

A	B
Name	Age
George	90
Maria	13
Ryan	22
Tim	98
George	61
Tim	68
Tim	50
Maria	91
George	16
Maria	23
Tim	60
Ryan	22
Maria	56
George	54
George	87
Ryan	33
Ryan	68
Ryan	45
George	21

=COUNTIF(range, criteria)  
=SUMIF(range, criteria, sum\_range)  
=AVERAGEIF(range, criteria, average\_range)

Which cells need to match your criteria?

Under what condition do I want to sum, count, or average?

Where are the values that I want to sum or average?

`COUNTIF(B2:B20,22) = 2`

`SUMIF(A2:A20,"Ryan",B2:B20) = 190`

`SUMIF(A2:A20,"<>Tim",B2:B20) = 702`

`AVERAGEIF(A2:A20,"Maria",B2:B20) = 45.75`

**COUNTIFS**, **SUMIFS**, and **AVERAGEIFS** are used when you want to evaluate a count, sum, or average based on multiple conditions or criteria

=COUNTIFS(criteria\_range1, criteria1, criteria\_range2 , criteria2...)

=SUMIFS(sum\_range, criteria\_range1, criteria1, criteria\_range2 , criteria2...)

=AVERAGEIFS(average\_range, criteria\_range1, criteria1, criteria\_range2 , criteria2...)

	A	B	C	D
1	Month	Tactic	Campaign	Clicks
2	Jan	Search	Google	166
3	Jan	Search	MSN	263
4	Jan	Display	Contextual	289
5	Jan	Display	Retargeting	137
6	Feb	Search	Google	124
7	Feb	Search	MSN	311
8	Feb	Display	Contextual	350
9	Feb	Display	Retargeting	384
10	Mar	Search	Google	168
11	Mar	Search	MSN	358
12	Mar	Display	Contextual	347
13	Mar	Display	Retargeting	390

COUNTIFS(B2:B13, "Search", D2:D13, ">200") = 3  
 SUMIFS(D2:D13, A2:A13, "Feb", B2:B13, "Display") = 734  
 AVERAGEIFS(D2:D13, A2:A13, "Jan", C2:C13, "MSN") = 263



#### PRO TIP:

If you use < or >, you need to add quotation marks as you would with text (i.e. ">200")

## EXAMPLE

### TEAM SALARY DASHBOARD

#### STEP 1

#### GO TO TEAM SALARY DASHBOARD SHEET

#### TEAM

1. COPY TEAM TO NEW SHEET
2. GET UNIQUE VALUES
  1. DATA TAB
    1. REMOVE DUPLICATE

#### GO BACK TEAM SALARY

- . SELECT C2
- a. DATA
- b. DATA VALIDATION
- c. ALLOW -> LIST
- d. SELECT SOURCE
- e. SHEET 2 (ALL DATA)



- Total Salary
- =SUMIFS('Salary Data'!\$E:\$E,'Salary Data'!\$B:\$B,'Team Salary Dashboard'!\$C\$2,'Salary Data'!\$A:\$A,'Team Salary Dashboard'!\$B5)
- Players
- =COUNTIFS('Salary Data'!\$B:\$B,'Team Salary Dashboard'!\$C\$2,'Salary Data'!\$A:\$A,'Team Salary Dashboard'!\$B5)
- Average Salary
- =C5/D5

## Module 4 – Lookup / Reference Functions

Objectives.

At the end of the module the participant is expected to:

1. Create Named Arrays in Excel
2. Apply Vlookup and Hlookup
3. Apply Row, Rows and Column, Columns
4. Apply Index, Match and Offset

Topics:

1. Named Arrays
2. Vlookup / Hlookup
3. Row, Rows/Column, Columns
4. Index, Match
5. Offset

## Named Arrays

Using **Named Arrays** can simplify a lookup function if you use the same data array in multiple formulas

For example, if you name the array from A1:D6 "Apparel"...

	A	B	C	D
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

...you can write your vlookup formula in either of the following ways:

=VLOOKUP(A1,\$A\$1:\$D\$6,2)  
=VLOOKUP(A1,Apparel,2)

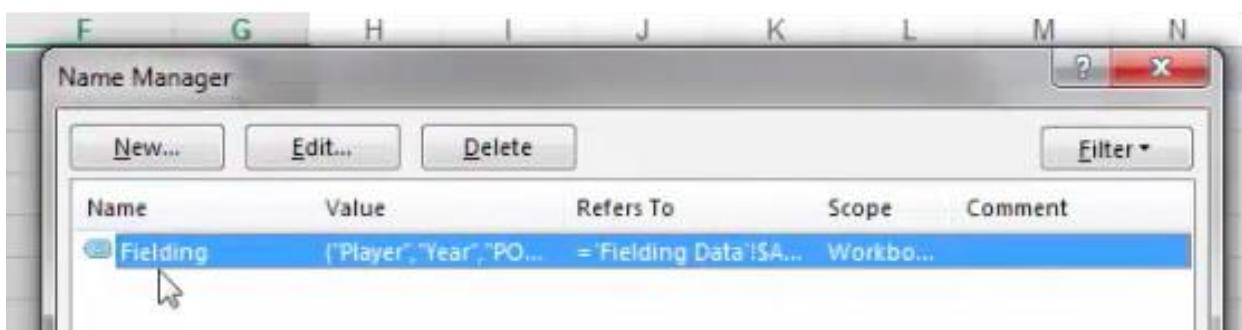
## EXAMPLE

- Open the DEMO\_MODULE4.XLSX file
- Goto Fielding data sheet
- SELECT ALL DATA
- Type Fielding (NAME BOX) (NAME OF THE ARRAY SET)

	A	B	C	D	E	F	G
12034	youngch04	2014	RF	6	0	1	0
12035	youngch04	2014	RF	2	0	0	0
12036	youngde03	2014	LF	29	3	0	0
12037	youngde03	2014	OF	32	4	0	1
12038	youngde03	2014	RF	3	1	0	1
12039	younger03	2014	2B	3	2	0	0
12040	younger03	2014	CF	0	0	0	0
12041	younger03	2014	LF	128	6	1	1
12042	younger03	2014	OF	128	6	1	1
12043	zeidjo01	2014	P	1	7	0	1
12044	zieglbr01	2014	P	6	16	2	2
12045	zimmejo02	2014	P	13	19	1	0
12046	zimmery01	2014	1B	23	0	1	2
12047	zimmery01	2014	3B	18	37	3	4
12048	zimmery01	2014	LF	48	2	0	0
12049	zimmery01	2014	OF	48	2	0	0
12050	zobribe01	2014	2B	105	185	6	32
12051	zobribe01	2014	CF	12	0	0	0
12052	zobribe01	2014	LF	51	1	0	0
12053	zobribe01	2014	OF	101	2	0	0
12054	zobribe01	2014	RF	38	1	0	0
12055	zobribe01	2014	SS	16	57	2	12
12056	zuninmi01	2014	C	1010	84	5	5

### Check named array

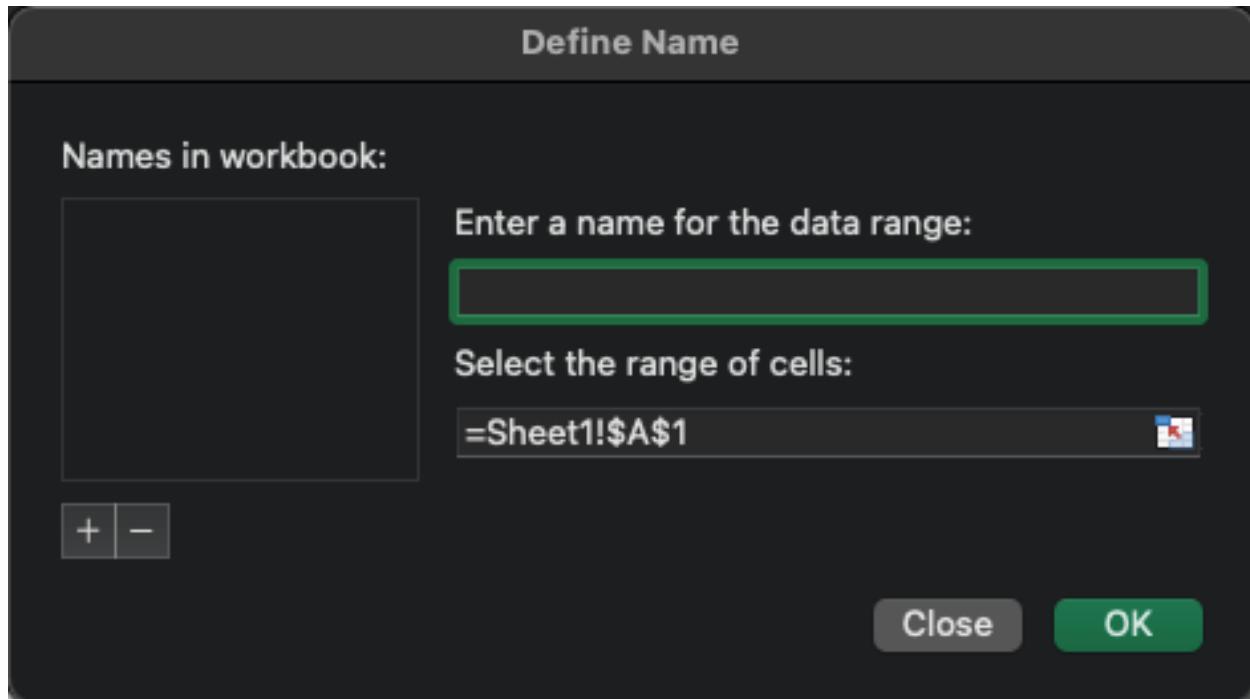
- Goto Formula tab
- Look for NAME MANAGER

Name	Value	Refers To	Scope	Comment
Fielding	("Player", "Year", "PO...	=Fielding Data!\$A...	Workbook...	

- Another option is to Goto the Formula tab
- Select DEFINE NAME
- Click New
  - Assign
  - Scope





MAC VERSION

## ROW/ROWS

The **ROW** function returns the row number of a given reference, while the **ROWS** function returns the number of rows in a given array or array formula

=ROW([reference])

=ROWS(array)

This example uses an array, which is why it includes the fancy {} signs – more on that in the ARRAY functions section

ROW(C10) = 10

ROWS(A10:D15) = 6

ROWS({1,2,3;4,5,6}) = 2

## Column / Columns

The **COLUMN** function returns the column number of a given reference, while the **COLUMNS** function returns the number of columns in a given array or array formula

=COLUMN([reference])

=COLUMNS(array)

### PRO TIP:



Leave the cell reference out and just write ROW() or COLUMN() to return the row or column number of the cell in which the formula is written

COLUMN(C10) = 3

COLUMNS(A10:D15) = 4

COLUMNS({1,2,3;4,5,6}) = 3

## EXAMPLE

- Open a new sheet
- Populate data using the RANDBETWEEN()

A	B	C	D	E	F	G
1	71	76	4			
2	52	19	7			
3	43	55	19			
4	64	88	42			
5	66	2	26			
6	42	62	77			
7	24	56	88			
8	65	99	77			
9	48	8	33			
10	80	89	31			
11	24	25	48			
12	82	6	76			
13						

<p>=ROW()</p>	<p>=COLUMN()</p>
---------------	------------------

<p>=ROW(B6)</p>	<p>=COLUMN(A3)</p>
<p>=ROWS(A1:C12)</p>	<p>=COLUMNS(A1:C12)</p>

## VLOOKUP

Let's take a look at one of Excel's most common reference functions – **VLOOKUP**:

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

This is the value that you are trying to match in the table array

This is where you are looking for the lookup value

Which column contains the data you're looking for?

Are you trying to match the exact lookup value (0), or something similar (1)?

A	B	C	D	
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

D2=VLOOKUP(A2, \$G\$1:\$H\$5, 2, 0)

To populate the Price in column D, we look up the name of the product in the data array from G1:H5 and return the value from the 2<sup>nd</sup> column over

## HLOOKUP

Use **HLOOKUP** if your table array is transposed (variables headers listed in rows)

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

This is the value that you are trying to match in the table array

This is where you are looking for the lookup value

Which column contains the data you're looking for?

Are you trying to match the exact lookup value (0), or something similar (1)?

A	B	C	D	
1	Product	Quantity	Product ID	Price
2	T-shirt	26	93754	\$14.99
3	Sweater	14	24783	\$49.99
4	Shorts	22	23984	\$24.50
5	Socks	36	58394	\$9.99
6	Spandex Unitard	2	27838	\$79.99

D2=HLOOKUP(A2, \$H\$1:\$L\$2, 2, 0)

With an HLOOKUP, we search for the product name in F1:J2 and return the value from the 2<sup>nd</sup> row down

## LAWS OF LOOKUPS

There are **two key rules** that constrain **VLOOKUP** and **HLOOKUP** formulas:



1. The lookup value must be in the **first column** of a VLOOKUP table array or the **first row** of a HLOOKUP table array
2. Excel will always return the value from the **top most row** or **left most column** of a table array when multiple instances of the lookup value are present

### PRO TIP:

Avoid breaking Law #2 by identifying a "Key" that is common to both datasets and is unique for every row (NOTE: Keys often take the form of a concatenation of multiple fields)

## EXAMPLE

- Insert a new column at Hitting data (KEY column) HITTING DATA SHEET

HITTING DATA SHEET				
	A	B	C	D
1	Key	Player	Year	G
2	=B2&C2	aardsda01	2010	53

- MOVE TO Fielding Data (SHEET)

FIELDING DATA SHEET				
	A	B	C	D
1	Key	Player	Year	POS
2	=B2&C2	aardsda01	2010	P
3		ahardfe01	2010	D

O	P	Q	R
POS	Putouts	Assists	Errors

- Copy the **Year** column from the Field data to Hitting data
- Goto the POS COLUMN

```
=VLOOKUP(A2,'Fielding Data'!$A$1:$H$12056,4,0)
```

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

Year    POS    Putouts    Assists    Errors

- Update the POS Column, make A2 Absolute

```
=VLOOKUP($A2,'Fielding Data'!$A$1:$H$12056,4,0)
```

- Populate the remaining data

POS

```
=VLOOKUP($A2,'Fielding Data'!$A$2:$H$12056,4,0)
```

Putouts

```
=VLOOKUP($A2,'Fielding Data'!$A$2:$H$12056,5,0)
```

Assists

```
=VLOOKUP($A2,'Fielding Data'!$A$2:$H$12056,6,0)
```

Errors

```
=VLOOKUP($A2,'Fielding Data'!$A$2:$H$12056,7,0)
```

DP

```
=VLOOKUP($A2,'Fielding Data'!$A$2:$H$12056,8,0)
```

## FIXING SOME ERRORS

- Apply filter to the column headers

- Check out the POS column (THERE ARE ITEMS WITH NA VALUES)
- Update it by adding IFERROR FUNCTION

```
=IFERROR(VLOOKUP($A2,'Fielding Data'!$A$1:$H$12056,4,0),"Other")
```

	IFERROR(value, value_if_error)	H	I	J	K
	AB R H	2B	3B	HR	RBI

- NEXT FIX THE Putout COLUMN
- CHECK THE DATA (NA) Appears
- SELECT THE Putout COLUMN, check the SUM does not match

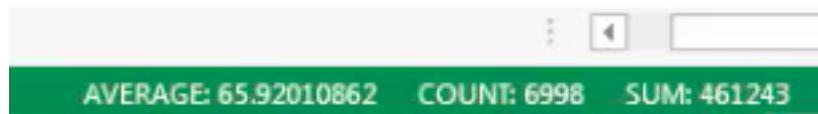


- FIX by Adding IFERROR

```
=IFERROR(VLOOKUP($A2,'Fielding Data'!$A$1:$H$12056,5,0),0)
```

	IFERROR(value, value_if_error)	H	I	J
	AB R H	2B	3B	HR

- Once FIX AVERAGE, COUNT, SUM will appear



- Add IFERROR to the remaining column
- =IFERROR(VLOOKUP(\$A2,'Fielding Data'!\$A\$2:\$H\$12056,6,0),0)
- =IFERROR(VLOOKUP(\$A2,'Fielding Data'!\$A\$2:\$H\$12056,7,0),0)
- =IFERROR(VLOOKUP(\$A2,'Fielding Data'!\$A\$2:\$H\$12056,8,0),0)

## INDEX

The **INDEX** function returns the value of a specific cell within an array

**=INDEX(array, row\_num, column\_num)**

What range of cells  
are you looking at?

How many rows down  
is the value you want?

How many columns over  
is the value you want?

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333
5	Wrench set	\$10.00	234
6	Chain Saw	\$55.48	23
7	Tool Box	\$19.99	5
8	Level	\$2.25	7

**INDEX(\$A\$1:\$C\$5, 5, 3) = 234**

In this case we're telling Excel to find the value of a cell somewhere within the array of A1:C5. Starting from the upper left, we move down to the 5<sup>th</sup> row and right to the 3<sup>rd</sup> column, to return the value of 234

## MATCH

The **MATCH** function returns the position of a specific value within a column or row

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

What value are you trying  
to find the position of?

In which row or column are  
you looking? (must be a 1-  
dimensional array)

Are you looking for the exact value  
(0), or anything close?

- 1: Find largest value < or = lookup\_value
- 0: Find exact lookup\_value
- 1: Find smallest value > or = lookup\_value

	A	B
1	Tools	Price
2	Hammer	\$5.00
3	Screw Driver	\$2.50
4	Pliers	\$3.34
5	Wrench set	\$10.00

**MATCH("Pliers", \$A\$1:\$A\$5, 0) = 4**

	A	B	C
1	Tools	Price	Inventory
2	Hammer	\$5.00	55
3	Screw Driver	\$2.50	66
4	Pliers	\$3.34	333

**MATCH(66, \$A\$3:\$C\$3, 0) = 3**

Matching the word "Pliers" in column A, we  
find it in the 4<sup>th</sup> row. Matching the number 66  
in row 3, we find it in the 3<sup>rd</sup> column

## INDEX / MATCH

INDEX and MATCH are commonly used in tandem to act like a LOOKUP function; the only difference is that INDEX/MATCH can find values in any column or row in an array

Example: Price Checker

	A	B	C	D
1		Small	Medium	Large
2	Sweater	\$10	\$12	\$15
3	Jacket	\$30	\$35	\$40
4	Pants	\$25	\$30	\$35
5				
6	Product:	Pants		
8	Size:	Medium		
10	PRICE:	?		

In this example, we want to populate the price of a given product and size in cell B10 by returning a particular value within the array B2:D4

**B10=INDEX(B2:D4, MATCH(B6,A2:A4,0), MATCH(B8,B1:D1,0))**

The number of rows down to index  
depends on what product I'm  
looking for, so we use a MATCH  
function and search for the value  
in cell B6 (in this case "Pants")

The number of columns over to  
index depends on what size I'm  
looking for, so we use a MATCH  
function and search for the value  
in cell B8 (in this case, "Medium")

Considering the output of each MATCH function, the formula is just a simple INDEX:

**B10 = INDEX(B2:D4, 3, 2) = \$30**

## EXAMPLE

- PRICE CHECKER EXAMPLE

	XSmall	Small	Medium	Large	XLarge
Socks	\$4.00	\$4.25	\$4.50	\$4.75	\$5.00
Shorts	\$12.00	\$12.50	\$13.00	\$13.50	\$14.00
Pants	\$24.00	\$25.00	\$26.00	\$27.00	\$28.00
T-Shirt	\$15.00	\$15.50	\$16.00	\$16.50	\$17.00
Sweater	\$30.00	\$31.00	\$32.00	\$33.00	\$34.00

Product:      Size:      =      Price

- STEP 1
  - CREATE A LIST UNDER PRODUCTS

	XSmall	Small	Medium	Large	XLarge
Socks	\$4.00	\$4.25	\$4.50	\$4.75	\$5.00
Shorts	\$12.00	\$12.50	\$13.00	\$13.50	\$14.00
Pants	\$24.00	\$25.00	\$26.00	\$27.00	\$28.00
T-Shirt	\$15.00	\$15.50	\$16.00	\$16.50	\$17.00
Sweater	\$30.00	\$31.00	\$32.00	\$33.00	\$34.00

Product:      Size:      =      Price

Data Validation  
=SBS3:SBS7

- SIZE

	XSmall	Small	Medium	Large	XLarge
Socks	\$4.00	\$4.25	\$4.50	\$4.75	\$5.00
Shorts	\$12.00	\$12.50	\$13.00	\$13.50	\$14.00
Pants	\$24.00	\$25.00	\$26.00	\$27.00	\$28.00
T-Shirt	\$15.00	\$15.50	\$16.00	\$16.50	\$17.00
Sweater	\$30.00	\$31.00	\$32.00	\$33.00	\$34.00

Product:      Size:      =      Price

Data Validation  
Settings      Input Message      Error Alert

Validation criteria

Allow: List      Ignore blank      In-cell dropdown

Data: between      Source: =SC\$2:\$G\$2

Apply these changes to all other cells with the same settings

Clear All      OK      Cancel

Cell formula bar: =INDEX(\$C\$3:\$G\$7,MATCH(B10,B3:B7,0),MATCH(C10,C2:G2,0))

A	B	C	D	E	F	G	H	I
				XSmall	Small	Medium	Large	XLarge
Socks	\$4.00	\$4.25	\$4.50	\$4.75	\$5.00			
Shorts	\$12.00	\$12.50	\$13.00	\$13.50	\$14.00			
Pants	\$24.00	\$25.00	\$26.00	\$27.00	\$28.00			
T-Shirt	\$15.00	\$15.50	\$16.00	\$16.50	\$17.00			
Sweater	\$30.00	\$31.00	\$32.00	\$33.00	\$34.00			

Product: Socks      Size: Small      =      Price: =INDEX(\$C\$3:\$G\$7,MATCH(B10,B3:B7,0),MATCH(C10,C2:G2,0))