MICROSOFT EXCEL:

FORMULAS & FUNCTIONS

★★★★★ *With Best-Selling Excel instructor **Chris Dutton***

MAVEN ANALYTICS

COURSE OUTLINE

1	**Excel Formulas 101**	*Syntax, reference types, errors, auditing, shortcuts, etc.*
2	**Conditionals & Logical Operators**	*IF, AND, OR, NOT, ISERROR, ISNUMBER, etc.*
3	**Basic Statistical Functions**	*MAX/MIN, RANK, RAND(), SUMIFS/COUNTIFS, SUMPRODUCT, etc.*
4	**Lookup/Reference Functions**	*VLOOKUP/HLOOKUP, INDEX/MATCH, OFFSET, etc.*
5	**Text Formulas**	*TEXT/VALUE, LEFT/MID/RIGHT, SEARCH, TRIM, LEN, etc.*
6	**Date & Time Functions**	*DATEVALUE, TODAY/NOW, DATEDIF, YEARFRAC, EOMONTH, etc.*
7	**Formula-Based Formatting**	*Creating, editing, and managing formula-driven formatting rules*
8	**Array Formulas**	*Vertical/Horizontal arrays, double unary, TRANSPOSE, etc.*
9	**Extra Bonus Functions**	*HYPERLINK, INDIRECT, WEBSERVICE, FILTERXML, etc.*

SETTING EXPECTATIONS

1. We'll be using Excel for **Windows/PC** (Excel 2013-2019)
   - *What you see on your screen* ***will not always match*** *mine, especially if you're using an older version of Excel*

2. We'll focus on many of Excel's most **powerful, widely-used** functions
   - *Excel's formula library includes nearly* ***500 functions****; we won't cover some of the more specialized categories (like Financial or Engineering functions), or those which require knowledge outside the scope of this course*

3. The goal of this course is to help you **master the building blocks**
   - *The beauty of Excel formulas is that no matter how complex they get, they're ALL comprised of simple pieces. We'll start by mastering each individual component before combining them in more complex ways*

4. Feeling stuck? **We've got your back.**
   - *If you have questions about the course material, feel free to post a question and we'll be happy to help*
   - *For project-specific questions, recommend posting to the* ***answers.microsoft.com*** *community forum*

FORMULAS 101

Formula Syntax

*All formulas start with an **equals sign***

*Arguments are always surrounded by **parentheses***

*Arguments are separated by **commas** in the US, but other regions may use different list separators (like **semi-colons**)*

= MATCH(lookup_value, lookup_array, [match_type])

The **function name** tells Excel what type of operation you're about to perform *(Excel offers ~500 functions)*

Note: Function names aren't case-sensitive, and aren't always required; basic arithmetic and logical operations often don't need one:

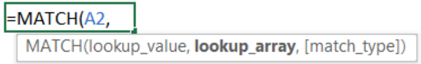
- *= A1 + B1*
- *= A1 / B1*
- *= A1 > B1*
- *= A1 = B1*

These are **arguments**, which vary by function and provide Excel with the info needed to evaluate a result

Note: Not all arguments are required; optional arguments are surrounded by square brackets *(like **[match_type]** above)*

Most functions have at least one required argument, but some don't require any, like **ROW()**, **COLUMN()**, **TODAY()** or **NOW()**

PRO TIP:



=MATCH(A2,

MATCH(lookup_value, **lookup_array**, [match_type])

*As you begin writing a formula, the **Function ScreenTips** box will guide you through each individual argument – this is an extremely helpful tool!*

Reference Types

Reference types allow you to "recycle" formulas across multiple cells, without having to manually update your references (which would be completely impractical)

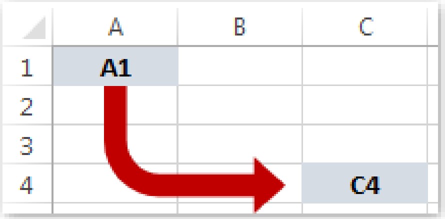
Cell references are **relative** by default (A1). This **allows the reference to change** as the formula is copied to new cells

The **$** symbol is used to create **fixed** references. You can fix entire cells ($A$1) or just the column ($A1) or row (A$1), which **prevents references from changing** as the formula is copied to new cells

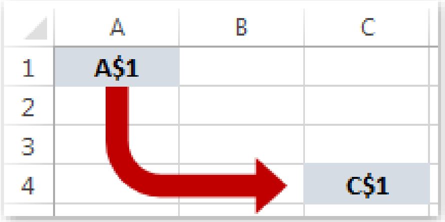
PRO TIP:

*Mastering reference types is my **#1 tip** for working efficiently with formulas*

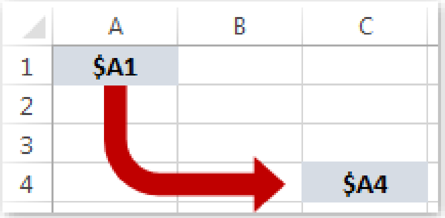
Relative Column & Row



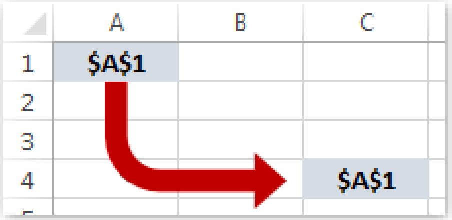
Relative Column, Fixed Row



Fixed Column, Relative Row



Fixed Column & Row

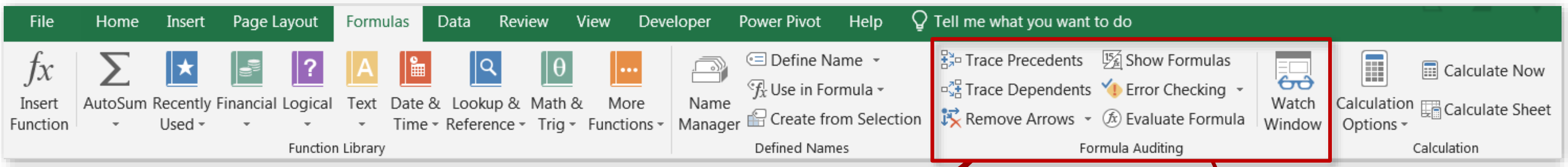


MAVEN ANALYTICS

Common Errors

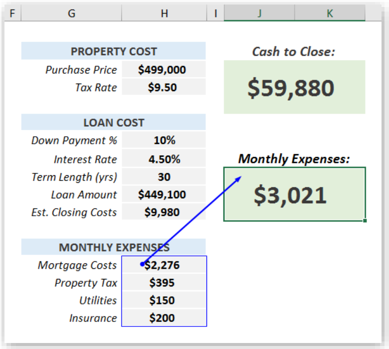
Error Type	What it means	How to fix it
######	**Column isn't wide enough to display values**	*Drag or double-click column border to increase width, or right-click to set custom column width*
#NAME?	**Excel does not recognize text in a formula**	*Make sure that function names are correct, references are 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*
#N/A	**Formula can't find a referenced value**	*Check that all references and formula arguments evaluate properly (the most common cause is a lookup value with no match)*

Auditing Tools



Trace Precedents

Identifies cells which **affect** the value of the one selected



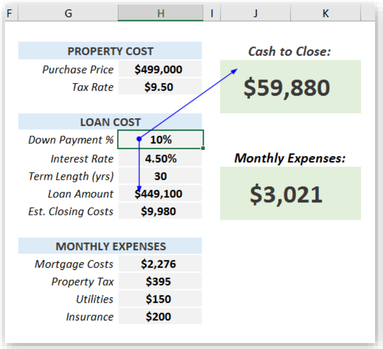
PROPERTY COST	
Purchase Price	$499,000
Tax Rate	$9.50
LOAN COST	
Down Payment %	10%
Interest Rate	4.50%
Term Length (yrs)	30
Loan Amount	$449,100
Est. Closing Costs	$9,980
MONTHLY EXPENSES	
Mortgage Costs	$2,276
Property Tax	$395
Utilities	$150
Insurance	$200

Cash to Close: $59,880

Monthly Expenses: $3,021

Trace Dependents

Identifies cells which **are affected by** the value of the one selected



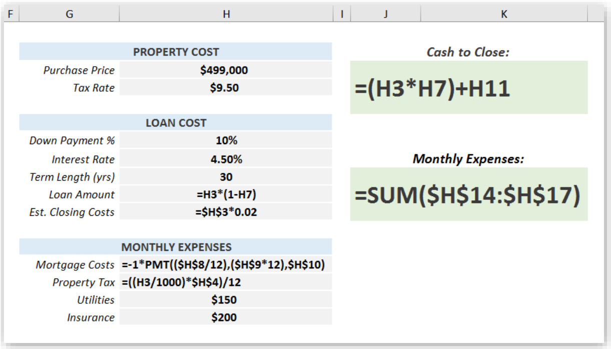
PROPERTY COST	
Purchase Price	$499,000
Tax Rate	$9.50
LOAN COST	
Down Payment %	10%
Interest Rate	4.50%
Term Length (yrs)	30
Loan Amount	$449,100
Est. Closing Costs	$9,980
MONTHLY EXPENSES	
Mortgage Costs	$2,276
Property Tax	$395
Utilities	$150
Insurance	$200

Cash to Close: $59,880

Monthly Expenses: $3,021

Show Formulas

Temporarily displays all formulas in the worksheet as **text strings**

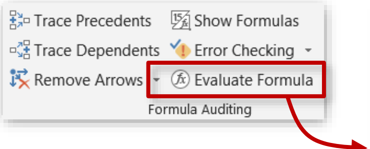


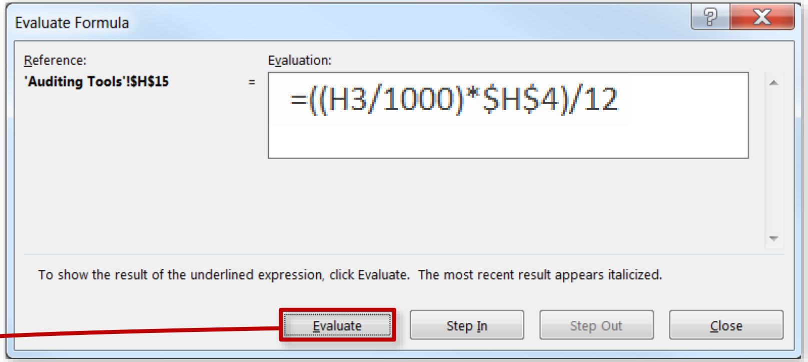
PROPERTY COST	
Purchase Price	$499,000
Tax Rate	$9.50
LOAN COST	
Down Payment %	10%
Interest Rate	4.50%
Term Length (yrs)	30
Loan Amount	=H3*(1-H7)
Est. Closing Costs	=$H$3*0.02
MONTHLY EXPENSES	
Mortgage Costs	=-1*PMT(($H$8/12),($H$9*12),$H$10)
Property Tax	=((H3/1000)*$H$4)/12
Utilities	$150
Insurance	$200

Cash to Close: =(H3*H7)+H11

Monthly Expenses: =SUM($H$14:$H$17)

TIP**: To <u>select</u> all cells containing formulas, use **Ctrl-G** to launch the Go-To menu, then select **Special > Formulas



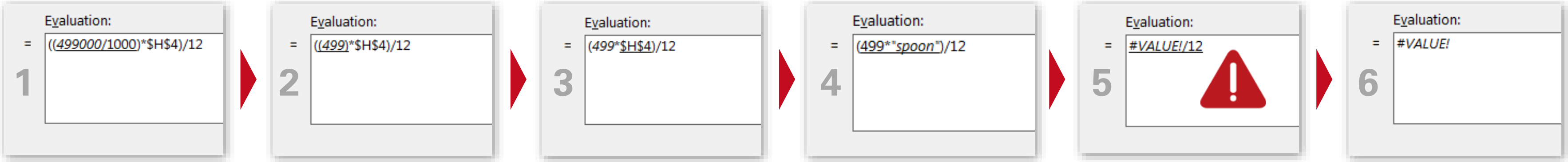


Reference: 'Auditing Tools'!$H$15 = Evaluation: =((H3/1000)*$H$4)/12

To show the result of the underlined expression, click Evaluate. The most recent result appears italicized.

Evaluate Formula

Allows you to cycle through each individual calculation step within a formula, see how each component evaluates, and pinpoint the source of the error



1 Evaluation: = ((499000/1000)*$H$4)/12

2 Evaluation: = ((499)*$H$4)/12

3 Evaluation: = (499*$H$4)/12

4 Evaluation: = (499*"spoon")/12

5 Evaluation: = #VALUE!/12

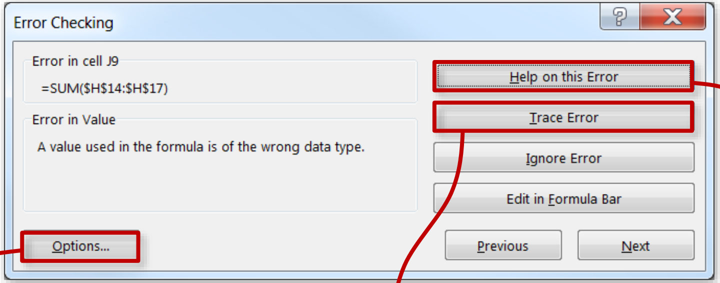
6 Evaluation: = #VALUE!

PRO TIP:

*Evaluate Formula is my **go-to tool** for breaking down complex or unfamiliar formulas*

Error Checking

Scans the sheet for errors and provides a summary with options to trace the source, ignore the error, modify your options, or link out to Microsoft support



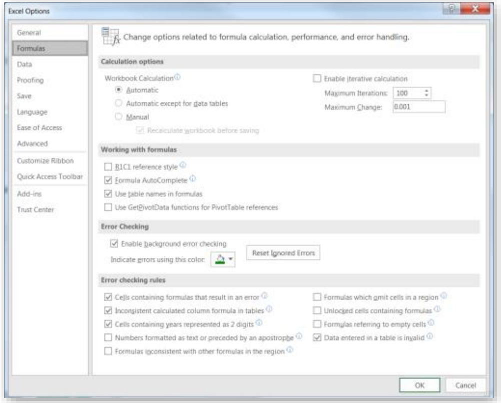
Error Checking

Error in cell J9

=SUM($H$14:$H$17)

Error in Value

A value used in the formula is of the wrong data type.



Excel Options

Change options related to formula calculation, performance, and error handling.

Calculation options

Workbook Calculation
- ● Automatic
- ○ Automatic except for data tables
- ○ Manual
- ☑ Recalculate workbook before saving

☐ Enable iterative calculation

Maximum Iterations: 100

Maximum Change: 0.001

Working with formulas

- ☐ R1C1 reference style
- ☑ Formula AutoComplete
- ☑ Use table names in formulas
- ☐ Use GetPivotData functions for PivotTable references

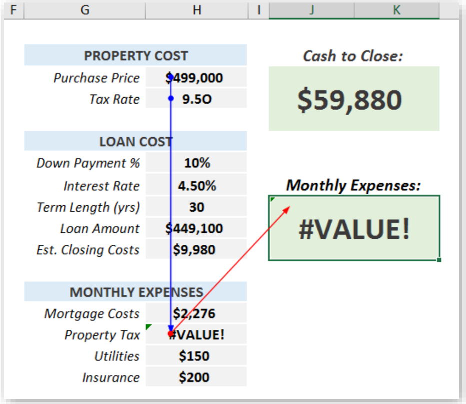
Error Checking

- ☑ Enable background error checking

Indicate errors using this color:

Error checking rules

- ☑ Cells containing formulas that result in an error
- ☑ Inconsistent calculated column formula in tables
- ☑ Cells containing years represented as 2 digits
- ☐ Numbers formatted as text or preceded by an apostrophe
- ☐ Formulas inconsistent with other formulas in the region
- ☐ Formulas which omit cells in a region
- ☐ Unlocked cells containing formulas
- ☐ Formulas referring to empty cells
- ☑ Data entered in a table is invalid



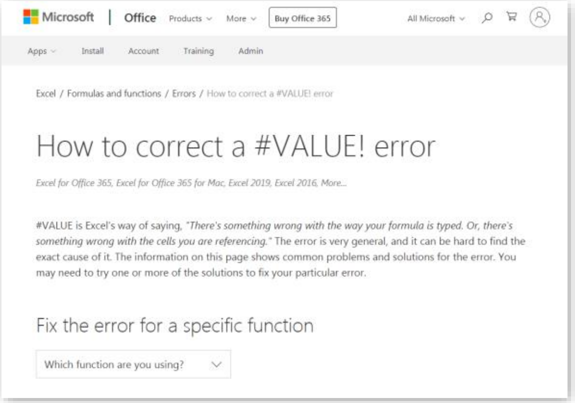
G	H
PROPERTY COST	
Purchase Price	$499,000
Tax Rate	9.50
LOAN COST	
Down Payment %	10%
Interest Rate	4.50%
Term Length (yrs)	30
Loan Amount	$449,100
Est. Closing Costs	$9,980
MONTHLY EXPENSES	
Mortgage Costs	$2,276
Property Tax	#VALUE!
Utilities	$150
Insurance	$200

Cash to Close:

$59,880

Monthly Expenses:

#VALUE!



Excel / Formulas and functions / Errors / How to correct a #VALUE! error

How to correct a #VALUE! error

Excel for Office 365, Excel for Office 365 for Mac, Excel 2019, Excel 2016, More...

#VALUE is Excel's way of saying, *"There's something wrong with the way your formula is typed. Or, there's something wrong with the cells you are referencing."* The error is very general, and it can be hard to find the exact cause of it. The information on this page shows common problems and solutions for the error. You may need to try one or more of the solutions to fix your particular error.

Fix the error for a specific function

Which function are you using?

CTRL Shortcuts

Ctrl + Arrow

- **Jumps to the last cell** in a data region, in the direction of the arrow

Ctrl + Shift + Arrow

- **Selects to the last cell** in a data region, in the direction of the arrow

Ctrl + Home/End

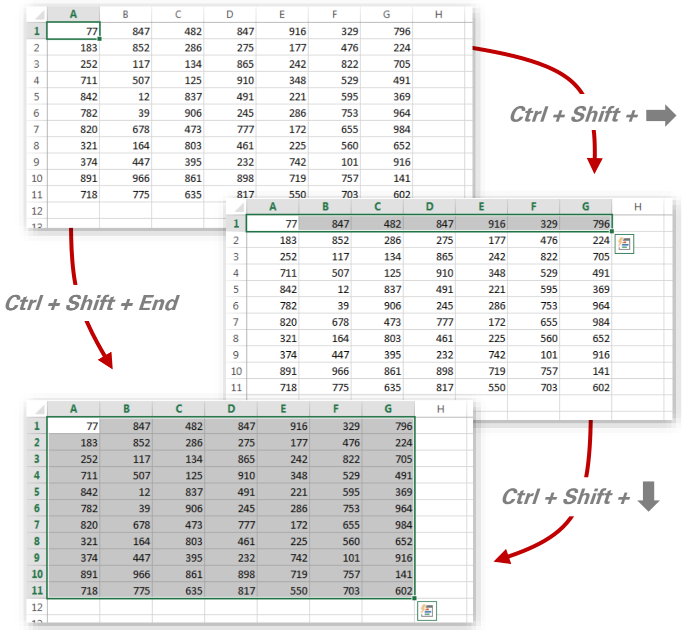
- Jumps to the **Home** (*top-left*) or **End** (*bottom-right*) cell in a region

Ctrl + .

- Jumps straight to **each corner** within a selected cell range

Ctrl + PgUp/PgDn

- **Switches worksheet tabs**, either to the **left** (*PgUp*) or **right** (*PgDn*)



	A	B	C	D	E	F	G	H
1	77	847	482	847	916	329	796	
2	183	852	286	275	177	476	224	
3	252	117	134	865	242	822	705	
4	711	507	125	910	348	529	491	
5	842	12	837	491	221	595	369	
6	782	39	906	245	286	753	964	
7	820	678	473	777	172	655	984	
8	321	164	803	461	225	560	652	
9	374	447	395	232	742	101	916	
10	891	966	861	898	719	757	141	
11	718	775	635	817	550	703	602	
12								

	A	B	C	D	E	F	G	H
1	77	847	482	847	916	329	796	
2	183	852	286	275	177	476	224	
3	252	117	134	865	242	822	705	
4	711	507	125	910	348	529	491	
5	842	12	837	491	221	595	369	
6	782	39	906	245	286	753	964	
7	820	678	473	777	172	655	984	
8	321	164	803	461	225	560	652	
9	374	447	395	232	742	101	916	
10	891	966	861	898	719	757	141	
11	718	775	635	817	550	703	602	

	A	B	C	D	E	F	G	H
1	77	847	482	847	916	329	796	
2	183	852	286	275	177	476	224	
3	252	117	134	865	242	822	705	
4	711	507	125	910	348	529	491	
5	842	12	837	491	221	595	369	
6	782	39	906	245	286	753	964	
7	820	678	473	777	172	655	984	
8	321	164	803	461	225	560	652	
9	374	447	395	232	742	101	916	
10	891	966	861	898	719	757	141	
11	718	775	635	817	550	703	602	
12								

FULL LIST: https://exceljet.net/keyboard-shortcuts

Fn Shortcuts

F1

- Launches the **Excel help** pane *(default)*
- Links to the **Microsoft Support** website *(tool-specific)*

F2

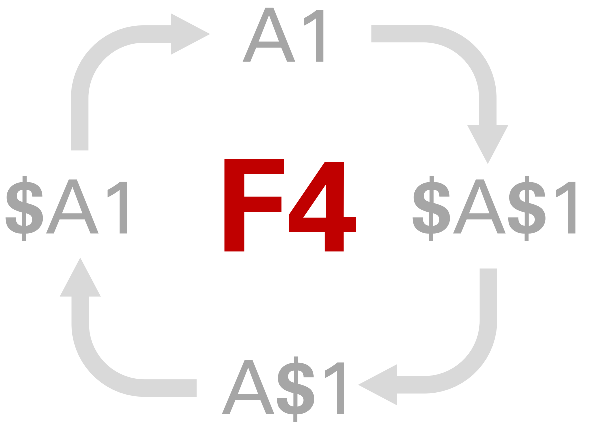
- Allows you to **edit** the active cell
- Highlights cells referenced by the active formula

F4

- Repeats the **last action** taken *(default)*
- Toggles **absolute/relative** cell references within a formula

F9

- Calculates all workbook formulas *(when in **manual** mode)*
- Evaluates **each function argument** within the formula bar



FULL LIST: https://exceljet.net/keyboard-shortcuts

Common Mac Shortcuts

Mac Shortcut	Purpose	PC Equivalent
Command-T	*Cycles between cell reference types*	**F4**
Command-Y	*Repeats the last user action*	**F4**
Control-U	*Displays cell ranges tied to a given formula*	**F2**
Command-Arrow	*Jumps to the edge of a contiguous data array*	**CTRL-ARROW**
Command-Shift-Arrow	*Extends a selection to the edge of a data array*	**CTRL-SHIFT-ARROW**
Command-Fn-Up/Down	*Jumps between workbook tabs*	**CTRL-PAGE UP/DOWN**

FULL LIST: https://exceljet.net/keyboard-shortcuts

Alt Key Tips

- Allow you to quickly access tools from ribbon menus and sub-menus using only the **keyboard** *(no clicks!)*
- Each keystroke takes you a layer deeper, until you land on the tool you need *(**Note**: Simply **press & release** the Alt key, instead of holding it down)*
- There are hundreds of combinations, so start by focusing on the tools that you use most frequently:

Some of my go-to key tips

Alt – H – V – V

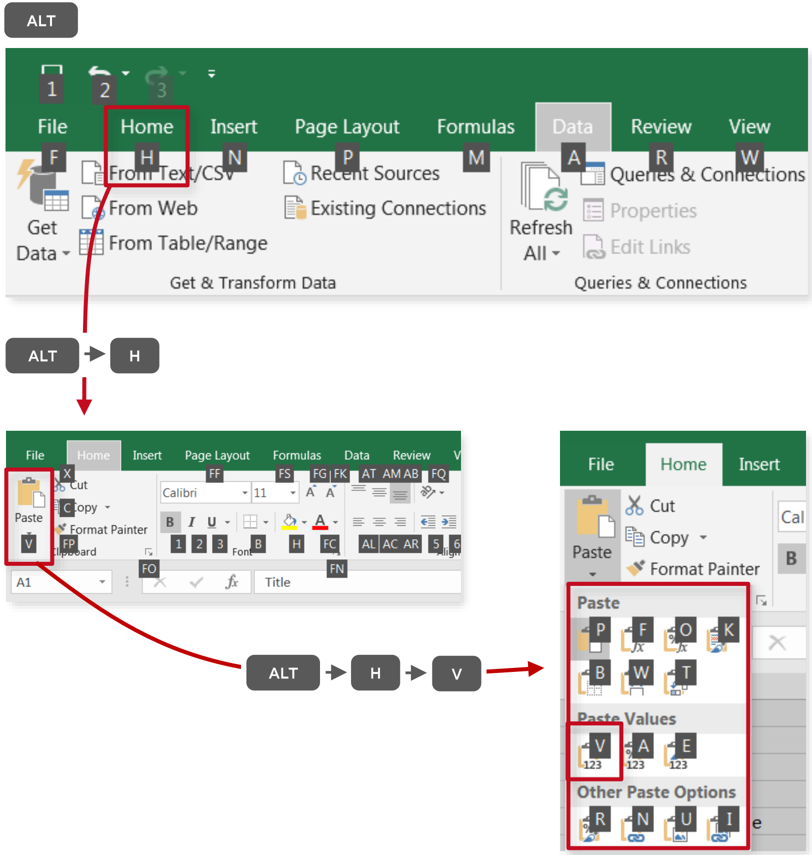
- Paste Special as Values

Alt – A – T

- Add or remove filters

Alt – M – V

- Evaluate Formula

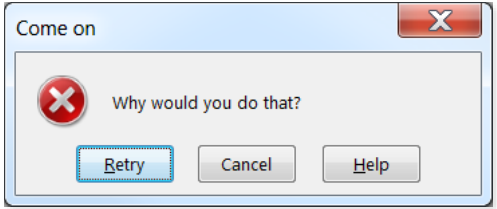


Data Validation

Restricts the values that a user can enter into a given cell, based on:

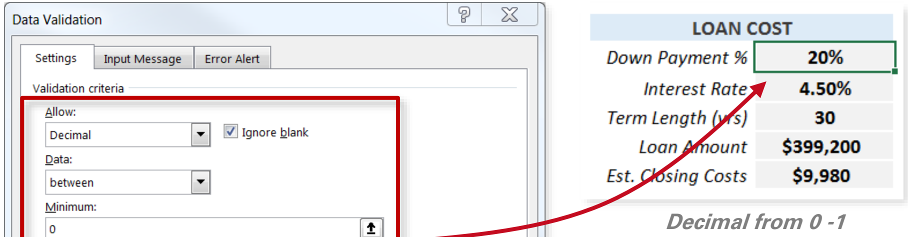
- **Number Type** *(Whole vs. Decimal)*
- **Value** *(Between, Less Than, Equal To, etc)*
- **List of Items** *(Based on cell range or manual list)*
- **Date/Time** *(Between, Less Than, Equal To, etc)*
- **Text Length** *(Between, Less Than, Equal To, etc)*
- **Custom** *(Formula-Driven)*

FUN FACT: You can customize your own **error alerts**!



Come on

Why would you do that?



Data Validation

Validation criteria

Allow: Decimal

☑ Ignore blank

Data: between

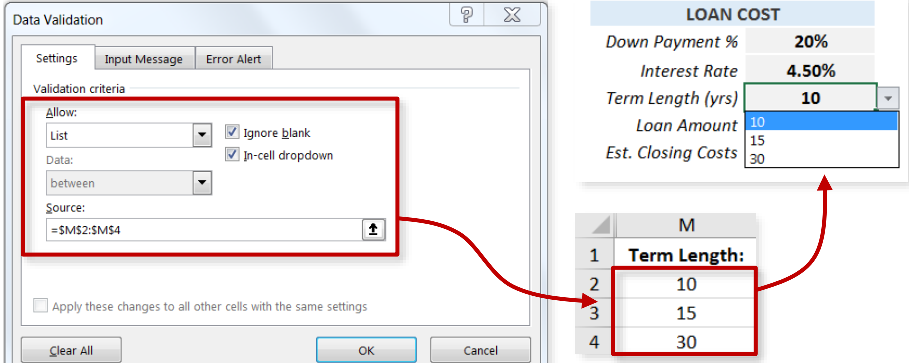
Minimum: 0

Maximum: 1

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

LOAN COST	
Down Payment %	**20%**
Interest Rate	**4.50%**
Term Length (yrs)	**30**
Loan Amount	**$399,200**
Est. Closing Costs	**$9,980**

Decimal from 0 -1



Data Validation

Validation criteria

Allow: List

☑ Ignore blank

☑ In-cell dropdown

Data: between

Source: =$M$2:$M$4

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

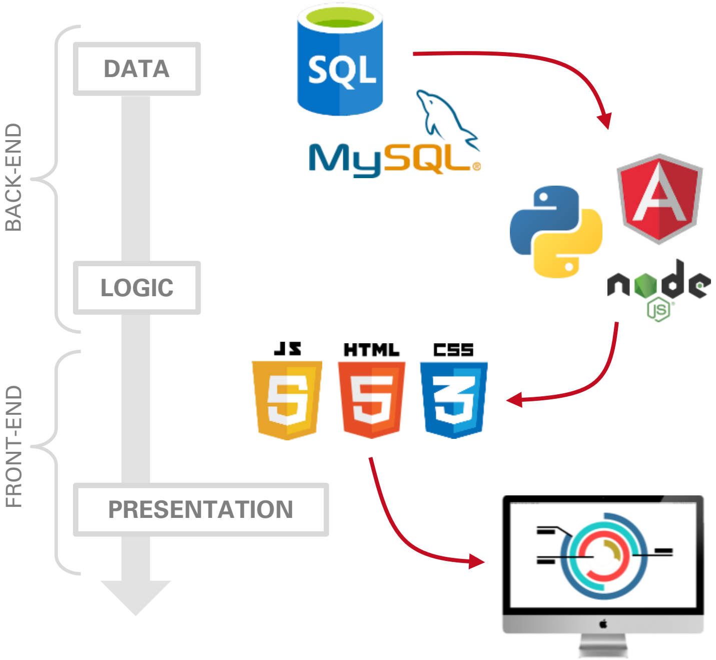
LOAN COST	
Down Payment %	**20%**
Interest Rate	**4.50%**
Term Length (yrs)	**10**
Loan Amount	10
	15
Est. Closing Costs	30

	M
1	**Term Length:**
2	10
3	15
4	30

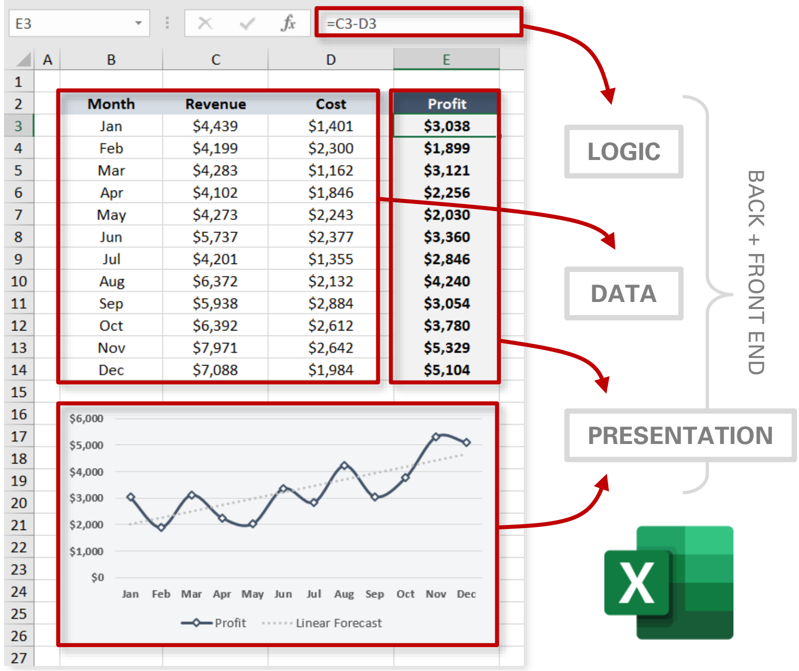
List of Items

Congrats, You're a Developer!

By definition, Excel is a **full-stack development** platform*; but rather than separating each layer of the process *(data, logic & presentation)*, Excel mixes them all within the same user interface:



VS



E3 | =C3-D3

Month	Revenue	Cost	Profit
Jan	$4,439	$1,401	$3,038
Feb	$4,199	$2,300	$1,899
Mar	$4,283	$1,162	$3,121
Apr	$4,102	$1,846	$2,256
May	$4,273	$2,243	$2,030
Jun	$5,737	$2,377	$3,360
Jul	$4,201	$1,355	$2,846
Aug	$6,372	$2,132	$4,240
Sep	$5,938	$2,884	$3,054
Oct	$6,392	$2,612	$3,780
Nov	$7,971	$2,642	$5,329
Dec	$7,088	$1,984	$5,104

This is NOT a claim that Excel is always the **right full-stack dev tool (or, in may cases, even a viable one). Rather, it's an effort to inspire users to think creatively about what Excel can do*

CONDITIONAL STATEMENTS

All **Conditional Statements** in Excel are based on simple "IF/THEN" statements:

*-**IF** it's raining, **THEN** bring an umbrella*
*-**IF** it's sunny, **THEN** bring sunglasses*
*-**IF** it's sunny **AND** it's summer, skip work and go to the beach*

You're basically saying ***"Hey Excel, if this statement is true, do this. Otherwise, do something else."***

IF Statements

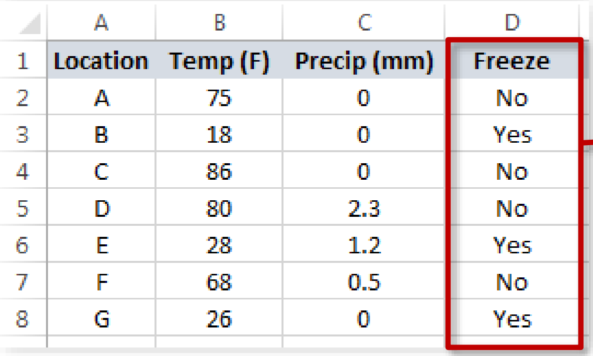
=IF(logical_test, [Value if True], [Value if False])

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**



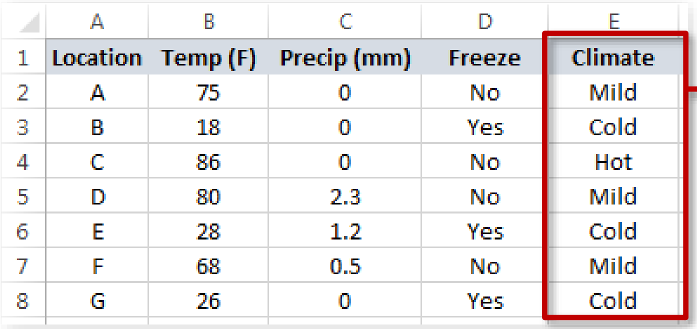
	A	B	C	D
1	Location	Temp (F)	Precip (mm)	Freeze
2	A	75	0	No
3	B	18	0	Yes
4	C	86	0	No
5	D	80	2.3	No
6	E	28	1.2	Yes
7	F	68	0.5	No
8	G	26	0	Yes

= IF(B2<=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"

Nested IF Statements

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



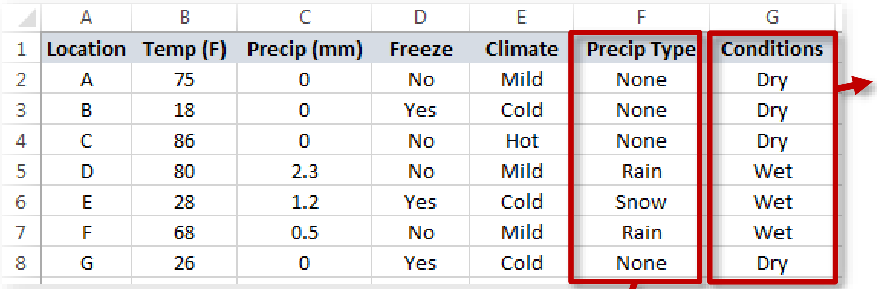
	A	B	C	D	E
1	Location	Temp (F)	Precip (mm)	Freeze	Climate
2	A	75	0	No	Mild
3	B	18	0	Yes	Cold
4	C	86	0	No	Hot
5	D	80	2.3	No	Mild
6	E	28	1.2	Yes	Cold
7	F	68	0.5	No	Mild
8	G	26	0	Yes	Cold

= IF(B2<40,"COLD",IF(B2>80,"HOT","MILD"))

If temp<40, climate = "Cold", if temp>80, climate = "Hot", otherwise climate = "Mild"

AND/OR Statements

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"))

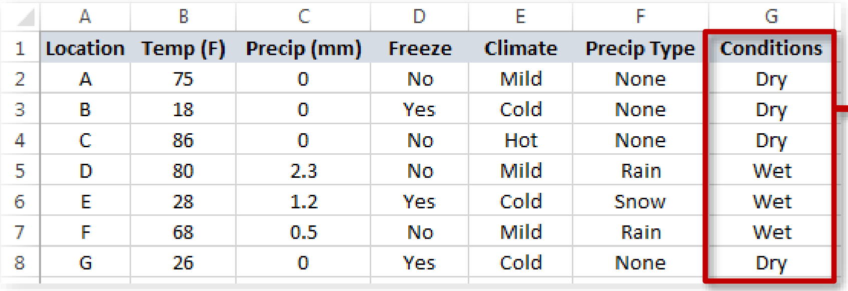
If the temp is below freezing AND the amount of precipitation > 0, then Precip Type = “Snow”, if the temp is above freezing AND the amount of precipitation >0, then Precip Type = “Rain”, otherwise Precip Type = “None”

PRO TIP:

When writing nested functions, copy/paste repetitive pieces and tweak individual elements to save time (rather than starting from scratch)

NOT/<> Operators

If you want to evaluate a case where a logical statement is *not* true, you can use either the NOT statement or a “<>” operator



	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(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

IFERROR

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

IS Statements

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*

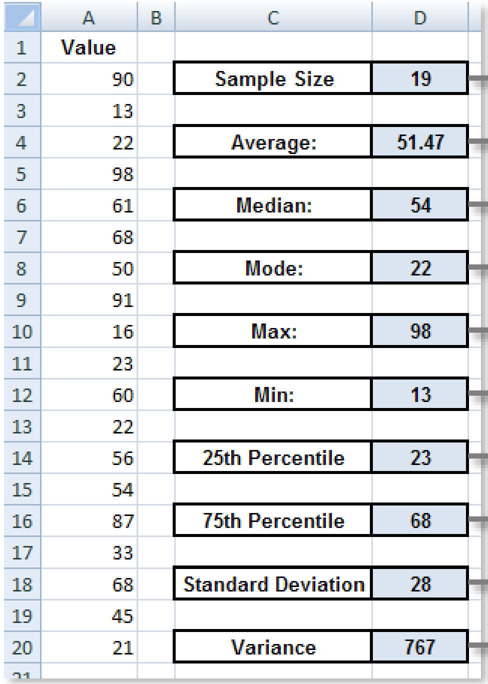
ISLOGICAL = *Checks whether the reference cell or value is a logical operator*

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

COMMON STATS FUNCTIONS

Basic Stats Functions

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	Value			
2	90		Sample Size	19
3	13			
4	22		Average:	51.47
5	98			
6	61		Median:	54
7	68			
8	50		Mode:	22
9	91			
10	16		Max:	98
11	23			
12	60		Min:	13
13	22			
14	56		25th Percentile	23
15	54			
16	87		75th Percentile	68
17	33			
18	68		Standard Deviation	28
19	45			
20	21		Variance	767
21				

=COUNT(A2:A20)

=AVERAGE(A2:A20)

=MEDIAN(A2:A20)

=MODE(A2:A20)

=MAX(A2:A20)

=MIN(A2:A20)

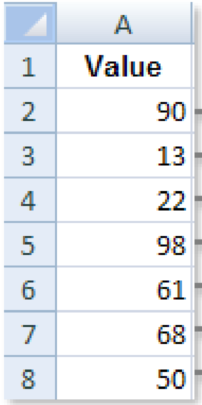
=PERCENTILE(A2:A20,.25)

=PERCENTILE(A2:A20,.75)

=STDEV(A2:A20)

=VAR(A2:A20)

RANK/SMALL/LARGE



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

RANK(A2,A2:A8) = 2

RANK(A3,A2:A8) = 7 *(lowest)*

RANK(A4,A2:A8) = 6

RANK(A5,A2:A8) = 1 *(highest)*

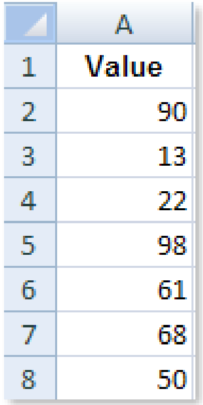
RANK(A6,A2:A8) = 4

RANK(A7,A2:A8) = 3

RANK(A8,A2:A8) = 5

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

The SMALL/LARGE functions return the nth smallest/largest values within an array



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

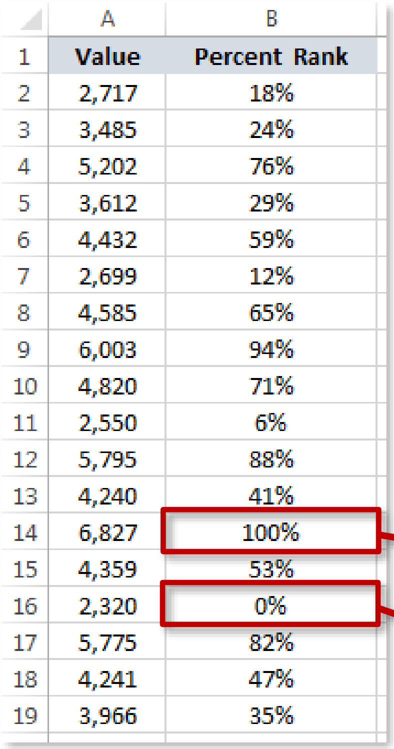
LARGE(A2:A8,2) = 90

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

SMALL(A2:A8,3) = 50

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

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?

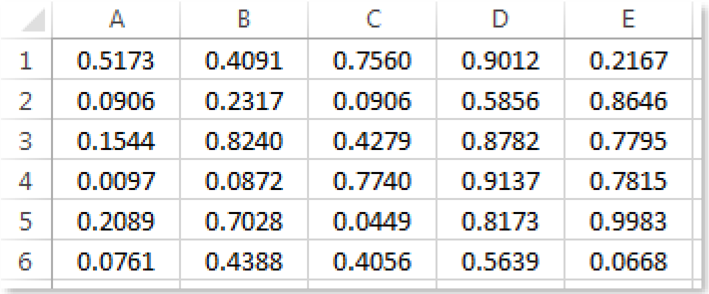
Which **value** within the range are you looking at?

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

PERCENTRANK($A$2:$A$19, A16) = 0% ***(lowest)***

RAND/RANDBETWEEN

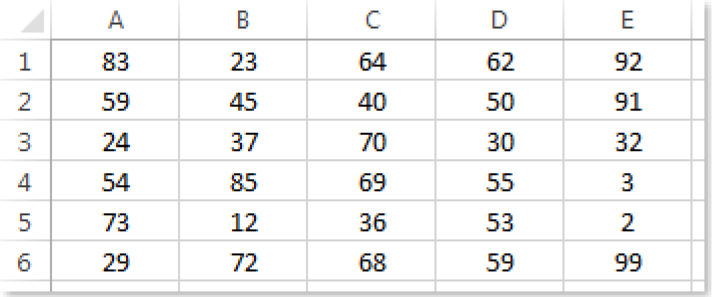
RAND() and RANDBETWEEN act like random number generators in Excel:



	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

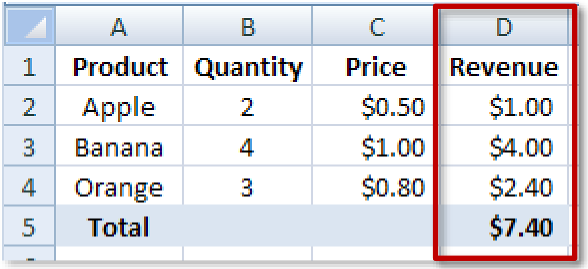
=RANDBETWEEN(0,100)

SUMPRODUCT

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

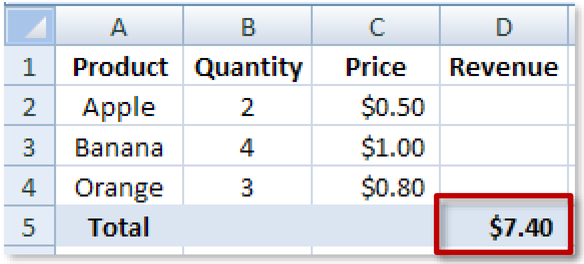
=SUMPRODUCT(array1, array2 … array_N)

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

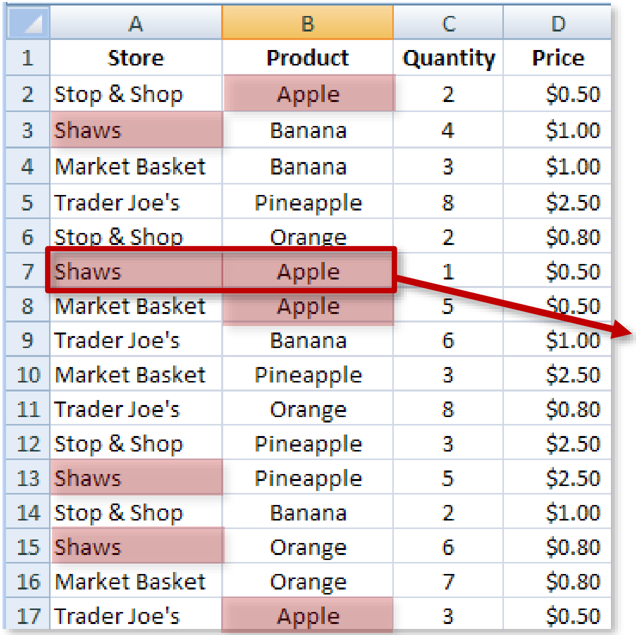
*Without using SUMPRODUCT, you could multiply quantity*price in each row and sum the products*



	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

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:

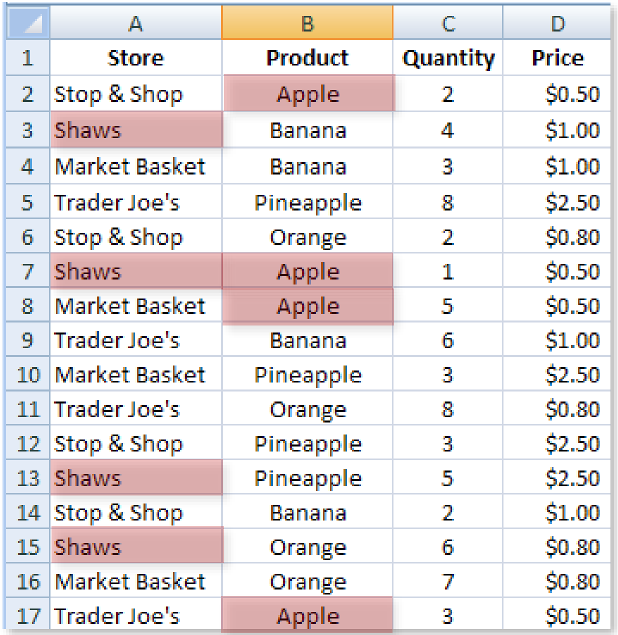
SUMPRODUCT((A2:A17="Shaws")*(B2:B17="Apple")*C2:C17*D2:D17) = $0.50

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?

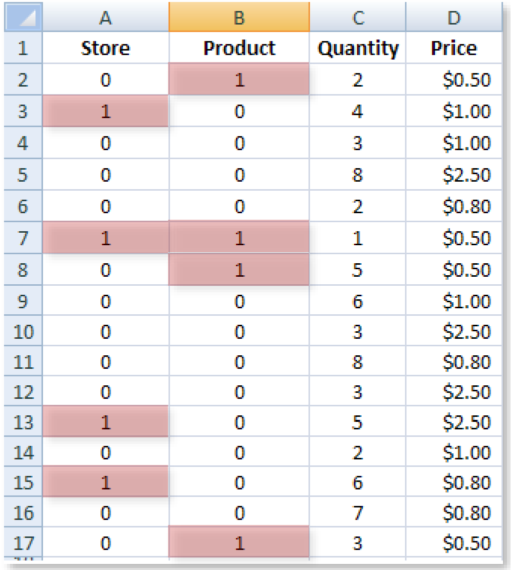
SUMPRODUCT((A2:A17="Shaws")*(B2:B17="Apple")*C2:C17*D2:D17) = $0.50



	A	B	C	D
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

What YOU see

What EXCEL sees



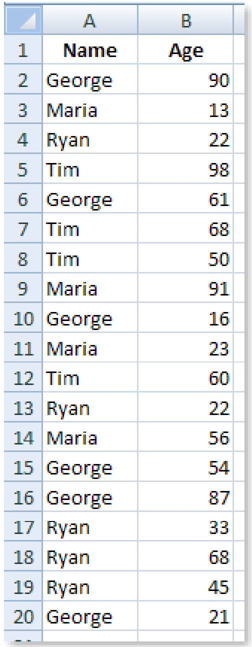
	A	B	C	D
1	Store	Product	Quantity	Price
2	0	1	2	$0.50
3	1	0	4	$1.00
4	0	0	3	$1.00
5	0	0	8	$2.50
6	0	0	2	$0.80
7	1	1	1	$0.50
8	0	1	5	$0.50
9	0	0	6	$1.00
10	0	0	3	$2.50
11	0	0	8	$0.80
12	0	0	3	$2.50
13	1	0	5	$2.50
14	0	0	2	$1.00
15	1	0	6	$0.80
16	0	0	7	$0.80
17	0	1	3	$0.50

*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

COUNTIF/SUMIF/AVERAGEIF

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



	A	B
1	Name	Age
2	George	90
3	Maria	13
4	Ryan	22
5	Tim	98
6	George	61
7	Tim	68
8	Tim	50
9	Maria	91
10	George	16
11	Maria	23
12	Tim	60
13	Ryan	22
14	Maria	56
15	George	54
16	George	87
17	Ryan	33
18	Ryan	68
19	Ryan	45
20	George	21

=COUNTIF(range, criteria)

=SUMIF(range, criteria, sum_range)

=AVERAGEIF(range, criteria, average_range)

range: Which cells need to match your criteria?

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

average_range: 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

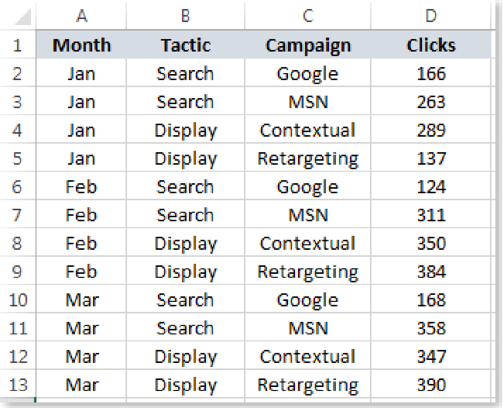
COUNTIFS/SUMIFS/AVERAGEIFS

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**

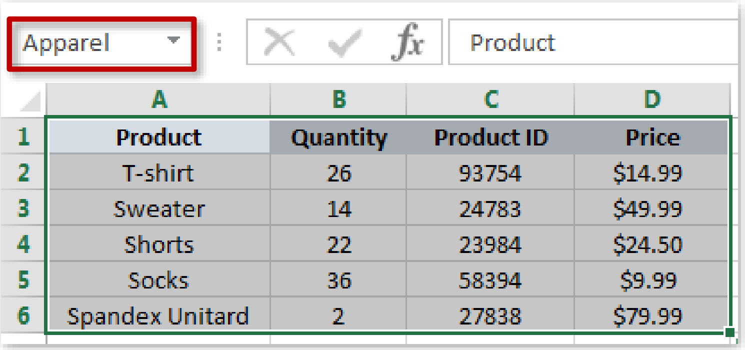
PRO TIP:

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

LOOKUP & REFERENCE FUNCTIONS

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"…



Apparel

Product

	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:

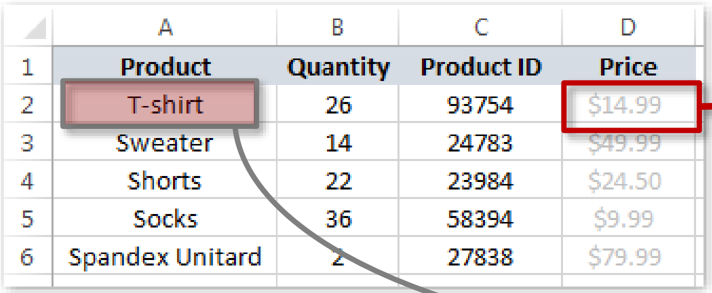
=VLOOKUP(A1,$A$1:$D$6,2)
=VLOOKUP(A1,Apparel,2)

VLOOKUP

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

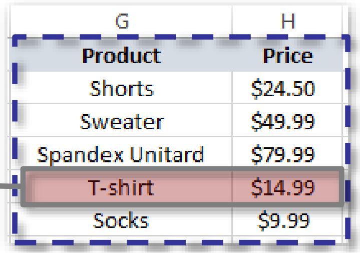
=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])

- lookup_value: This is the **value** that you are trying to match in the table array
- table_array: This is **where** you are looking for the lookup value
- col_index_num: **Which column** contains the data you’re looking for?
- [range_lookup]: 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)



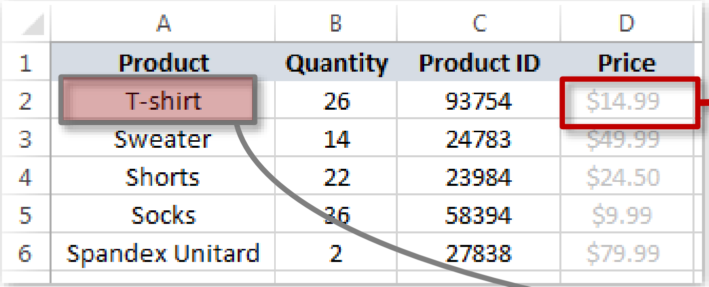
G	H
Product	Price
Shorts	$24.50
Sweater	$49.99
Spandex Unitard	$79.99
T-shirt	$14.99
Socks	$9.99

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 2nd column over

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

=HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])

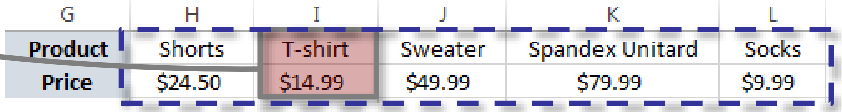
- lookup_value: This is the **value** that you are trying to match in the table array
- table_array: This is **where** you are looking for the lookup value
- row_index_num: **Which row** contains the data you're looking for?
- [range_lookup]: Are you trying to match the **exact** lookup value (0), or 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	26	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 2nd row down



G	H	I	J	K	L
Product	Shorts	T-shirt	Sweater	Spandex Unitard	Socks
Price	$24.50	$14.99	$49.99	$79.99	$9.99

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)

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)

ROW(C10) = 10

ROWS(A10:D15) = 6

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

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

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

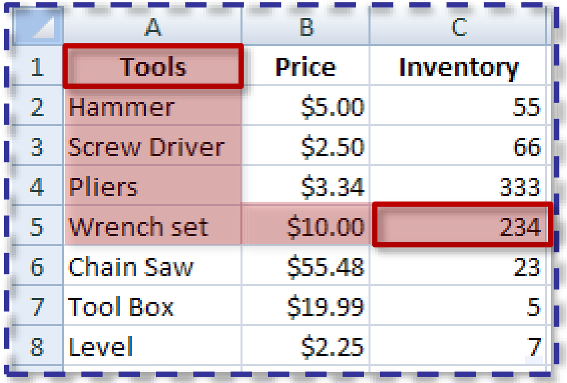
The **INDEX** function returns the *value* of a specific cell within an array

=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 **5th row** and right to the **3rd 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?

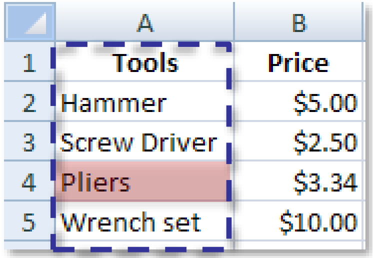
In which row or column are you looking? (**must be a 1-dimensional array**)

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

1: Find largest value < or = lookup_value

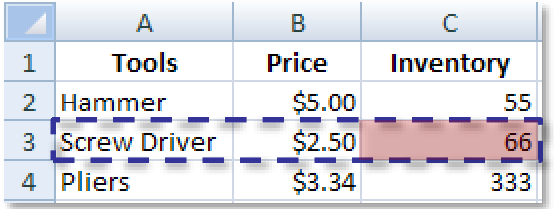
0: Find exact lookup_value

-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

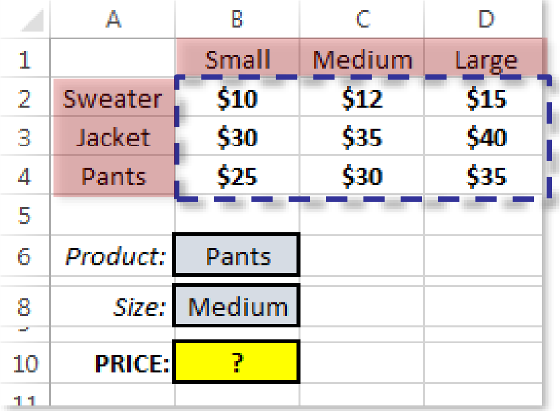
MATCH(66,$A$3:$C$3, 0) = 3

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

MAVEN ANALYTICS

XLOOKUP

XLOOKUP can retrieve values from a table or range by matching a lookup value, and offers more flexibility than **VLOOKUP**, **HLOOKUP**, or **INDEX & MATCH** formulas

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

- lookup_value: Which **value** are you looking to match?
- lookup_array: **Where** are you trying to find a match for your lookup value?
- return_array: Where are the **values** you want to retrieve?
- [if_not_found]: What if the lookup value **isn't found** in the lookup array?
- [match_mode]: Are you looking for an **exact**, **approximate**, or **wildcard** match?
- [search_mode]: Do you want to search **top down** or **bottom up**?

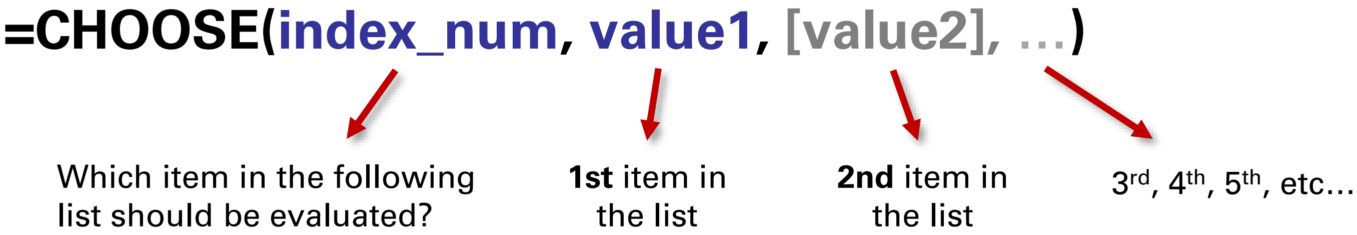
IMPORTANT NOTE: XLOOKUP is currently only available for **Office 365** subscribers

XLOOKUP VS. VLOOKUP

XLOOKUP	VLOOKUP
✓ Can retrieve a **dynamic array** of results	✗ Can only return a **single value**
✓ Can lookup values **anywhere** in an array (left or right, horizontal or vertical)	✗ Can only lookup values to the **right**, requires **HLOOKUP** for horizontal matching
✓ Defaults to **exact match**	✗ Defaults to **approximate match**
✓ Supports native **wildcard** text matching	✗ Does not natively support **wildcard** matching
✓ Includes **built-in error handling** when a lookup value is not found	✗ Requires an additional **IFERROR** function for error handling
✓ Can find approximate matches in **unsorted** lists	✗ Requires **sorted** lists for approximate matching
✓ Can search **top-down** or **bottom-up**	✗ Only searches **top-down**

CHOOSE

The CHOOSE function selects a value, cell reference, or function to perform from a list, based on a given index number



=CHOOSE(index_num, value1, [value2], …)

Which item in the following list should be evaluated?

1st item in the list

2nd item in the list

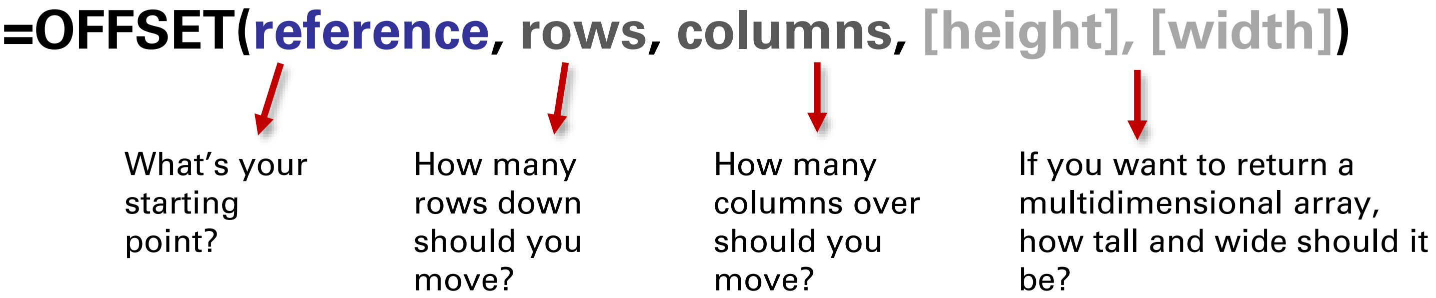
3rd, 4th, 5th, etc…

FUN FACTS ABOUT **CHOOSE:**

- List items can include **numbers**, **cell references**, **defined names**, **formulas**, or **text** *(or a mix!)*
- CHOOSE acts like an **INDIRECT** function, and can interpret cell references instead of treating them as text
- You can combine CHOOSE with other functions, or nest it directly into a cell reference

OFFSET

The OFFSET function is similar to INDEX, but can return either the value of a cell within an array (like INDEX) or a specific *range* of cells



=OFFSET(reference, rows, columns, [height], [width])

reference: What's your starting point?

rows: How many rows down should you move?

columns: How many columns over should you move?

[height], [width]: If you want to return a multidimensional array, how tall and wide should it be?

An OFFSET formula where [height]=1 and [width]=1 will operate exactly like an INDEX. A more common use of OFFSET is to create dynamic arrays (like the Scroll Chart example in the appendix)

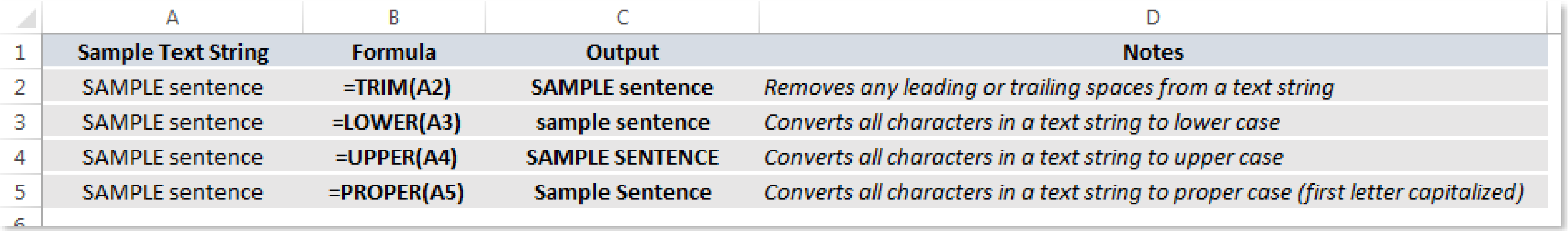
PRO TIP:

Don't use OFFSET or INDEX/MATCH when a simple VLOOKUP will do the trick

TEXT FUNCTIONS

TRIM/UPPER/LOWER/PROPER

Text functions can be used to standardize formatting, particularly the TRIM, UPPER, LOWER, and PROPER functions:



	A	B	C	D
1	**Sample Text String**	**Formula**	**Output**	**Notes**
2	SAMPLE sentence	**=TRIM(A2)**	**SAMPLE sentence**	*Removes any leading or trailing spaces from a text string*
3	SAMPLE sentence	**=LOWER(A3)**	**sample sentence**	*Converts all characters in a text string to lower case*
4	SAMPLE sentence	**=UPPER(A4)**	**SAMPLE SENTENCE**	*Converts all characters in a text string to upper case*
5	SAMPLE sentence	**=PROPER(A5)**	**Sample Sentence**	*Converts all characters in a text string to proper case (first letter capitalized)*
6				

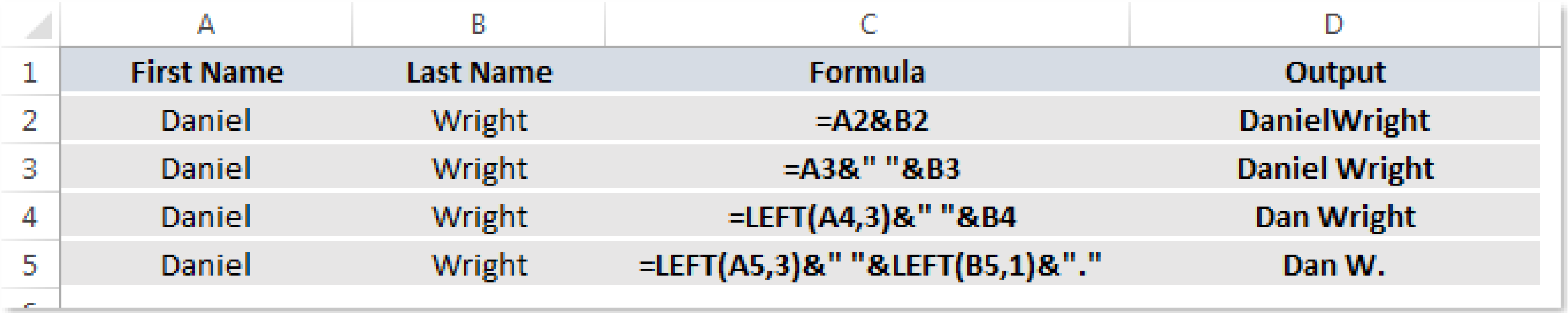
PRO TIP:

If two text strings are identical except one has a trailing space, they will look exactly the same but Excel will treat them as completely different values; TRIM will make them equivalent

CONCATENATE

CONCATENATE allows you to combine text, cell values, or formula outputs into a single text string

Note: Rather than typing **“=CONCATENATE(*Text1, Text2…*)”**, you can simply separate each piece of the resulting text string with an ampersand (“**&**”)



	A	B	C	D
1	First Name	Last Name	Formula	Output
2	Daniel	Wright	=A2&B2	DanielWright
3	Daniel	Wright	=A3&" "&B3	Daniel Wright
4	Daniel	Wright	=LEFT(A4,3)&" "&B4	Dan Wright
5	Daniel	Wright	=LEFT(A5,3)&" "&LEFT(B5,1)&"."	Dan W.

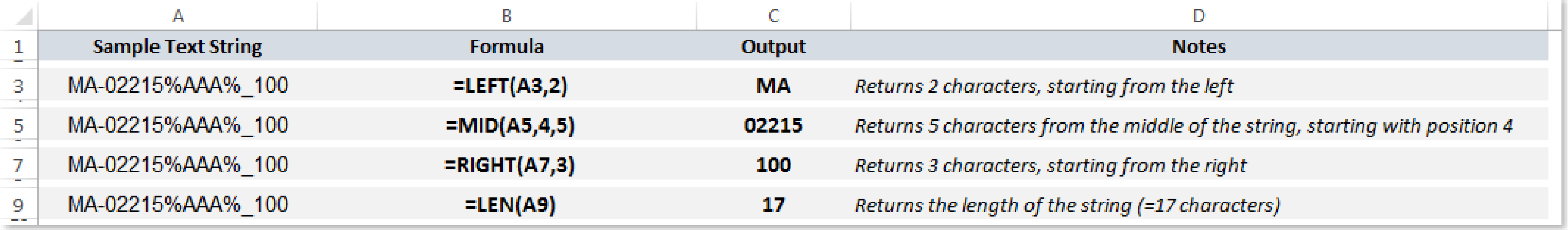
LEFT/MID/RIGHT/LEN

The LEFT, MID, and RIGHT functions return a specific number of characters from a location within a text string, and LEN returns the total number of characters

=LEFT(text, [num_chars])

=RIGHT(text, [num_chars])

=MID(text, start_num, num_chars)



	A	B	C	D
1	**Sample Text String**	**Formula**	**Output**	**Notes**
3	MA-02215%AAA%_100	**=LEFT(A3,2)**	**MA**	*Returns 2 characters, starting from the left*
5	MA-02215%AAA%_100	**=MID(A5,4,5)**	**02215**	*Returns 5 characters from the middle of the string, starting with position 4*
7	MA-02215%AAA%_100	**=RIGHT(A7,3)**	**100**	*Returns 3 characters, starting from the right*
9	MA-02215%AAA%_100	**=LEN(A9)**	**17**	*Returns the length of the string (=17 characters)*

MAVEN ANALYTICS

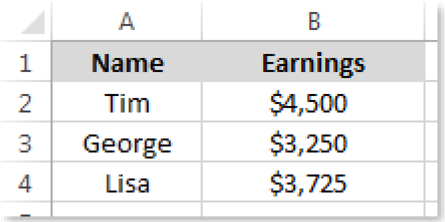
TEXT/VALUE

The TEXT function converts a numeric value to text and assigns a particular format

=TEXT(value, format_text)

Numeric value, formula that evaluates to a numeric value, or reference to a cell containing a numeric value

Numeric format as a text string enclosed in quotes (i.e. “**m/d/yyyy**”, “**$0.00**” or “**#,##0.00**”



	A	B
1	Name	Earnings
2	Tim	$4,500
3	George	$3,250
4	Lisa	$3,725

=“Lisa earned ”&B4 *returns* **“Lisa earned 3725”**

=“Lisa earned ”&TEXT(B4“$#,###”) *returns* **“Lisa earned $3,725”**

PRO TIP:

Use VALUE to convert a text string that represents a number into a value

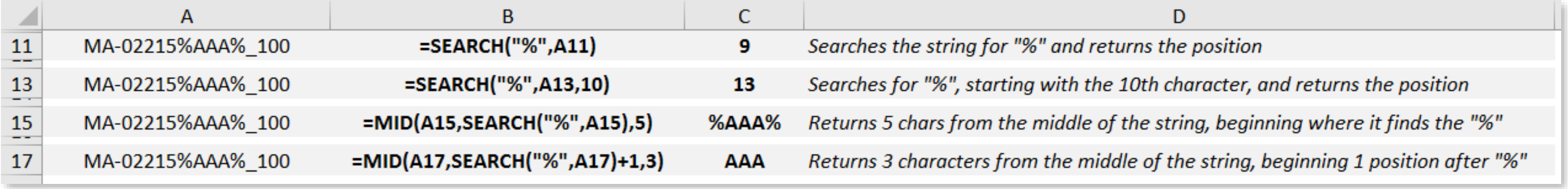
The SEARCH function returns the number of the character at which a specific character or text string is first found (otherwise returns #VALUE! error)

=SEARCH(find_text, within_text, [start_num])

find_text: What character or string are you searching for?

within_text: Where is the text that you’re searching through?

[start_num]: Search from the beginning (default) or after a certain number of characters?



	A	B	C	D
11	MA-02215%AAA%_100	=SEARCH("%",A11)	9	*Searches the string for "%" and returns the position*
13	MA-02215%AAA%_100	=SEARCH("%",A13,10)	13	*Searches for "%", starting with the 10th character, and returns the position*
15	MA-02215%AAA%_100	=MID(A15,SEARCH("%",A15),5)	%AAA%	*Returns 5 chars from the middle of the string, beginning where it finds the "%"*
17	MA-02215%AAA%_100	=MID(A17,SEARCH("%",A17)+1,3)	AAA	*Returns 3 characters from the middle of the string, beginning 1 position after "%"*

PRO TIP:

The FIND function works exactly the same way, but is case-sensitive

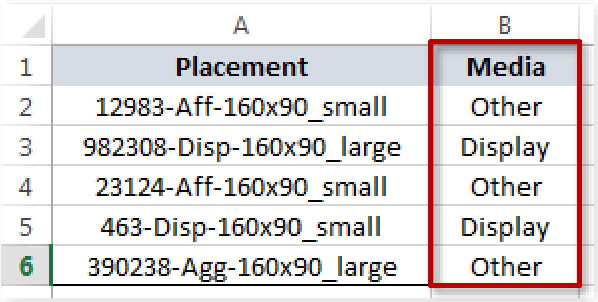
IF(ISNUMBER(SEARCH

IF(ISNUMBER(SEARCH is powerful combination of functions that can be used to classify data based on cells that contain specific strings of text

=IF(ISNUMBER(SEARCH(find_text, within_text)),value_if_true, value_if_false)

Searches for a specific string of text within a given cell

Returns one value if that string is found (TRUE), and another if it is not found (FALSE)



	A	B
1	Placement	Media
2	12983-Aff-160x90_small	Other
3	982308-Disp-160x90_large	Display
4	23124-Aff-160x90_small	Other
5	463-Disp-160x90_small	Display
6	390238-Agg-160x90_large	Other

=IF(ISNUMBER(SEARCH("Disp",A2)),"Display","Other")

Search the cells in column A for the text string "Disp" and classify column B as "Display" if you find it, "Other" if you don't

MAVEN ANALYTICS

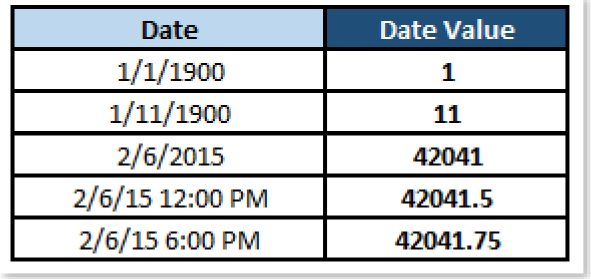
DATE & TIME FUNCTIONS

Date Value

Every date in Excel has an associated date value, which is how Excel calculates the passage of time (using midnight on 1/1/1900 as the starting point)

Excel recognizes most typed dates and automatically applies a common format (i.e. m/d/yyyy), along with an associated date value (cell format → General)

__Note:__ If you type a date in a format that Excel does NOT recognize, it will be treated as text and there will be no associated date value; however, you can use a __DATEVALUE__ or __TIMEVALUE__ function to convert unformatted dates or times into serial values



Date	Date Value
1/1/1900	1
1/11/1900	11
2/6/2015	42041
2/6/15 12:00 PM	42041.5
2/6/15 6:00 PM	42041.75

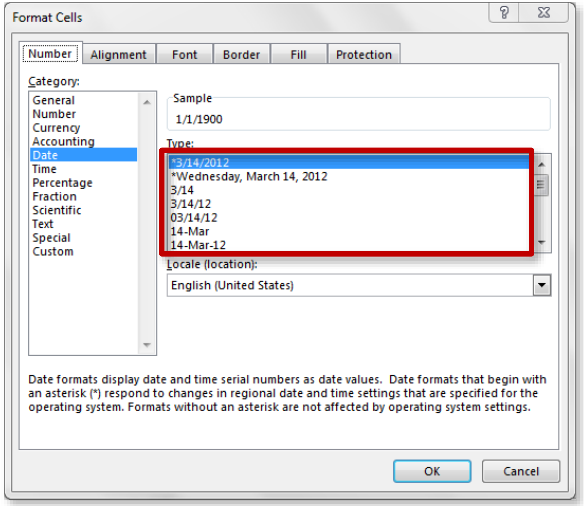
Jan 1,1900 is the first date with an assigned date value (1). Feb 6, 2015 is the 42,041st day since 1/1/1900, so its date value = 42041

Date values can also indicate fractions of days: 42041.5 translates to noon on 2/6/2015 (50% through the day), and 42041.75 translates to 6:00pm on 2/6/2015 (75% through the day)

Date Formatting

To format dates in Excel, you can either select a preset option from the “Date” category of the “Format Cells” dialog box, OR create your own custom format

Preset Formats:



Category:

- General
- Number
- Currency
- Accounting
- Date
- Time
- Percentage
- Fraction
- Scientific
- Text
- Special
- Custom

Sample

1/1/1900

Type:

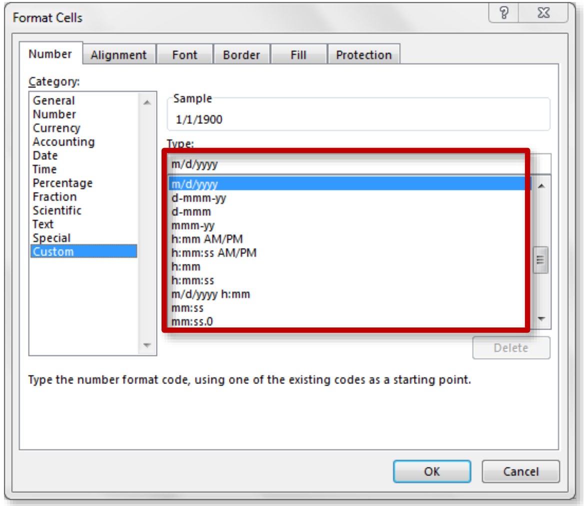
- *3/14/2012
- *Wednesday, March 14, 2012
- 3/14
- 3/14/12
- 03/14/12
- 14-Mar
- 14-Mar-12

Locale (location):

English (United States)

Date formats display date and time serial numbers as date values. Date formats that begin with an asterisk (*) respond to changes in regional date and time settings that are specified for the operating system. Formats without an asterisk are not affected by operating system settings.

Custom Format:



Category:

- General
- Number
- Currency
- Accounting
- Date
- Time
- Percentage
- Fraction
- Scientific
- Text
- Special
- Custom

Sample

1/1/1900

Type:

m/d/yyyy

- m/d/yyyy
- d-mmm-yy
- d-mmm
- mmm-yy
- h:mm AM/PM
- h:mm:ss AM/PM
- h:mm
- h:mm:ss
- m/d/yyyy h:mm
- mm:ss
- mm:ss.0

Type the number format code, using one of the existing codes as a starting point.

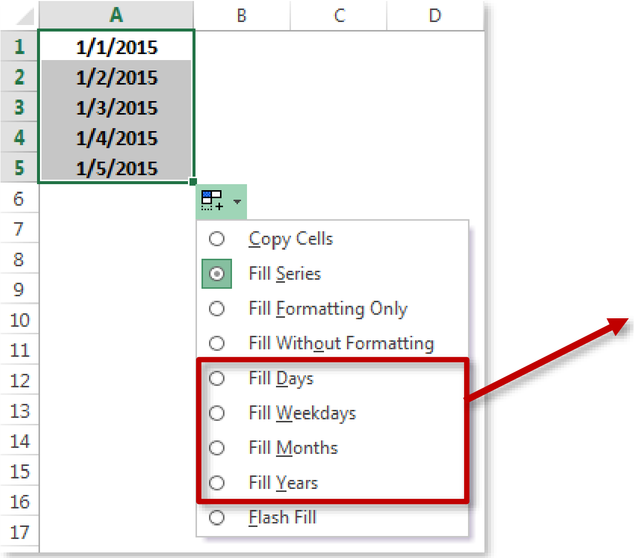
You can build your own custom formats using combinations of date/time codes. For example:

d = *day w/out leading zero (1-31)*
dd = *day w/ leading zero (01-31)*
ddd = *day-of-week (Sat)*
dddd = *day-of-week (Saturday)*
m = *month w/out leading zero (1-15)*
mm = *month w/ leading zero (01-15)*
mmm = *month abbreviation (Jan)*
mmmm = *full month (January)*
yy = *last 2 digits of year (15)*
yyyy = *full year (2015)*

(full list available at support.office.com)

Fill Series

When you drag the corner of a cell containing a date, Excel automatically applies subsequent values automatically using Fill Series options:



	A	B	C	D
1	1/1/2015			
2	1/2/2015			
3	1/3/2015			
4	1/4/2015			
5	1/5/2015			

- ○ Copy Cells
- ● Fill Series
- ○ Fill Formatting Only
- ○ Fill Without Formatting
- ○ Fill Days
- ○ Fill Weekdays
- ○ Fill Months
- ○ Fill Years
- ○ Flash Fill

*Click the **Auto Fill Options** button to determine exactly which values your subsequent cells should take:*

Copy Cells = Repeats the same value in all cells

Fill Days = Increases the date by 1 day per cell

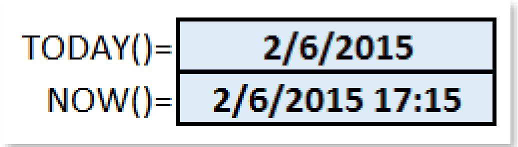
Fill Weekdays = Increases the date by 1 day per cell (excluding weekends)

Fill Months = Increases the date by 1 month per cell

Fill Years = Increases the date by 1 year per cell

The TODAY() and NOW() functions return the current date or exact time

***Note:** These are **volatile** functions, meaning that they change with every worksheet calculation*



TODAY()=	2/6/2015
NOW()=	2/6/2015 17:15

This is what the TODAY() and NOW() functions return at 5:15pm on February 6, 2015. Note that these values will automatically update with every change made to the workbook

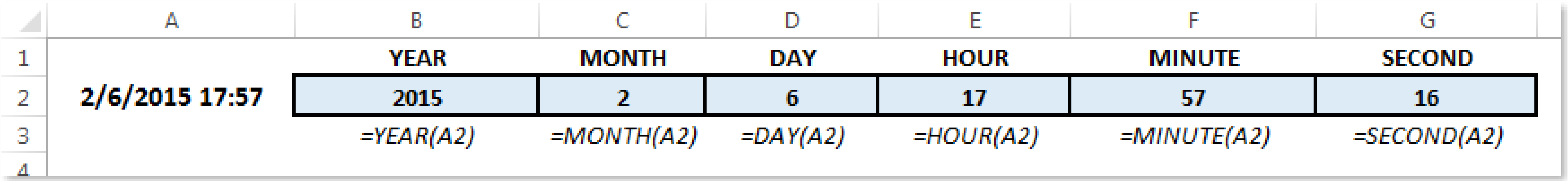
PRO TIP:

Make sure to enter TODAY() and NOW() functions with both parentheses included – these functions don't refer to other cells

Serialization Formulas

Excel will always calculate dates and times based on their *precise* underlying serial values, but what if you need to work with less-specific values, like months instead of days, or hours instead of seconds?

The YEAR, MONTH, DAY, HOUR, MINUTE, and SECOND functions extract individual components of a given date:



	A	B	C	D	E	F	G
1		YEAR	MONTH	DAY	HOUR	MINUTE	SECOND
2	2/6/2015 17:57	2015	2	6	17	57	16
3		=YEAR(A2)	=MONTH(A2)	=DAY(A2)	=HOUR(A2)	=MINUTE(A2)	=SECOND(A2)
4							

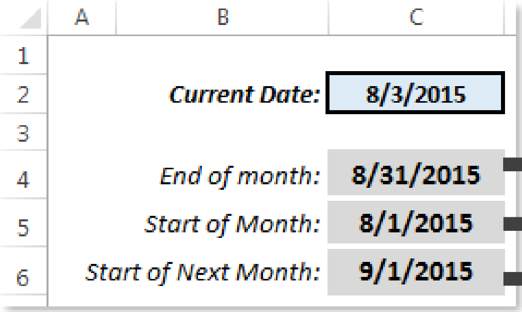
EOMONTH

Use the EOMONTH function to calculate the last day of a given month, or to calculate the start/end dates of previous or future months

=EOMONTH(start_date, months)

Reference to the cell containing the start/current date

Number of months before or after the start/current date (positive number yields a date in the future, negative number yields a date in the past



	A	B	C
1			
2		*Current Date:*	8/3/2015
3			
4		*End of month:*	8/31/2015
5		*Start of Month:*	8/1/2015
6		*Start of Next Month:*	9/1/2015

=EOMONTH(C2, 0)

=EOMONTH(C2, -1)+1

=EOMONTH(C2, 0)+1

YEARFRAC

YEARFRAC calculates the fraction of a year represented by the number of whole days between two dates

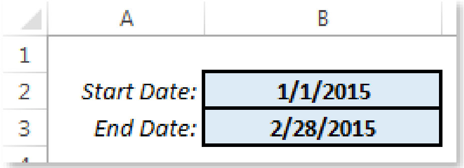
=YEARFRAC(start_date, end_date, [basis])

Reference to the cell containing the start date

Reference to the cell containing the end date

Option specify the type of day count to use:

- **0 (default)** = *US (NASD) 30/360*
- **1** = *actual/actual* ***(RECOMMENDED)***
- **2** = *actual/360*
- **3** = *actual/365*
- **4** = *European 30/360*



	A	B
1		
2	*Start Date:*	1/1/2015
3	*End Date:*	2/28/2015

=YEARFRAC(B2, B3, 1) = **15.9%**

=YEARFRAC(B2, B3, 2) = **16.1%**

PRO TIP:

YEARFRAC is a great tool for pacing and projection calculations

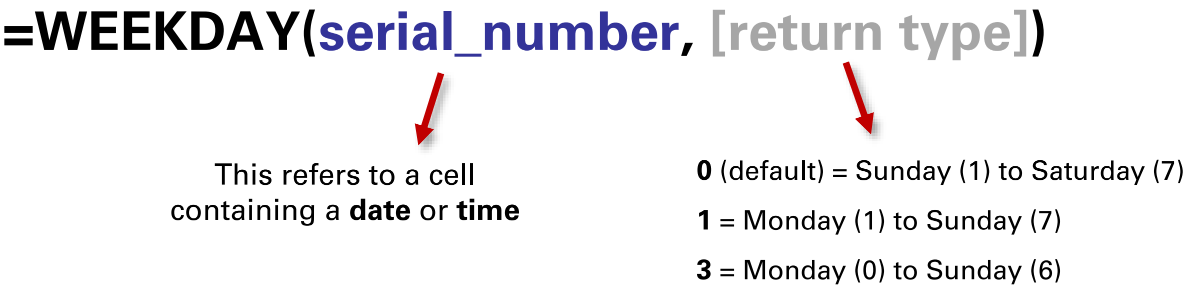
WEEKDAY

If you want to know which day of the week a given date falls on, there are two ways to do it:

1) Use a custom cell format of either “ddd” (Sat) or “dddd” (Saturday)

*-Note that this doesn’t change the underlying **value**, only how that value is displayed*

2) Use the **WEEKDAY** function to return a serial value corresponding to a particular day of the week (either 1-7 or 0-6)



=WEEKDAY(serial_number, [return type])

This refers to a cell containing a **date** or **time**

0 (default) = Sunday (1) to Saturday (7)

1 = Monday (1) to Sunday (7)

3 = Monday (0) to Sunday (6)

WORKDAY/NETWORKDAYS

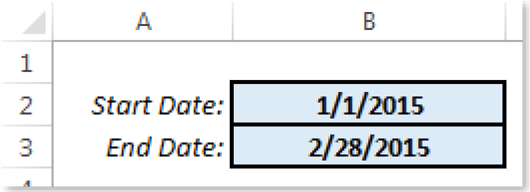
WORKDAY returns a date that is a specified number of days before or after a given start date, excluding weekends and (optionally) holidays; NETWORKDAYS counts the number of workdays between two dates:

=WORKDAY(start_date, days, [holidays])

- start_date: This refers to the cell containing the start date
- days: Number of days before or after start date
- [holidays]: Optional reference to a list of holiday dates

=NETWORKDAYS(start_date, end_date, [holidays])

- start_date: This refers to the cell containing the start date
- end_date: This refers to the cell containing the end date
- [holidays]: Optional reference to a list of holiday dates



	A	B
1		
2	*Start Date:*	1/1/2015
3	*End Date:*	2/28/2015

=WORKDAY(B2, 20) = 1/29/2015

=NETWORKDAYS(B2, B3) = 42

DATEDIF

DATEDIF calculates the number of days, months, or years between two dates

=DATEDIF(start_date, end_date, unit)

- start_date: Reference to the cell containing the start date
- end_date: Reference to the cell containing the end date
- unit: How do you want to calculate the difference?

"D" = *# of days between dates*

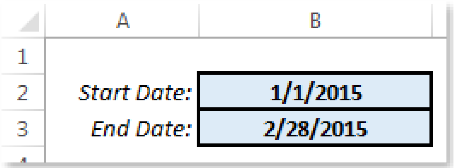
"M" = *# of months between dates*

"Y" = *# of years between dates*

"MD" = *# of days between dates, ignoring months and years*

"YD" = *# of days between dates, ignoring years*

"YM" = *# of months between dates, ignoring days and years*



	A	B
1		
2	*Start Date:*	1/1/2015
3	*End Date:*	2/28/2015

=DATEDIF(B2, B3, "D") = 58

=DATEDIF(B2, B3, "MD") = 27

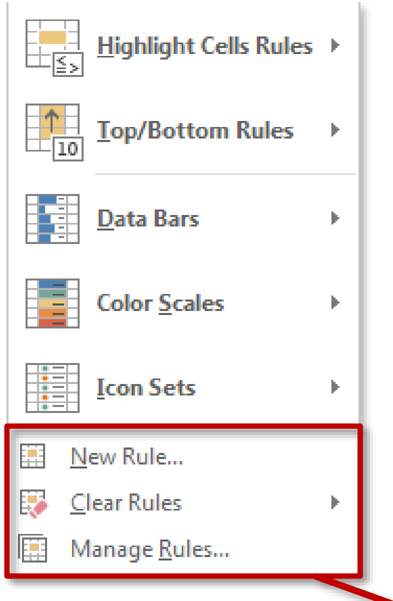
PRO TIP:

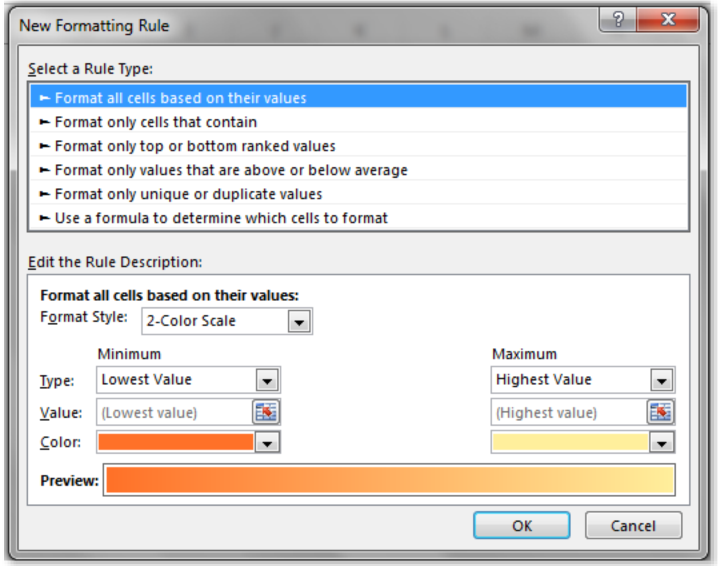
If you only need to calculate the # of days between dates, just use subtraction

FORMULA-BASED FORMATS

Formula-Based Formatting

If you want to go rogue, you can adjust the style of existing conditional formats or create your own **formula-based rules**





Select a Rule Type:

- Format all cells based on their values
- Format only cells that contain
- Format only top or bottom ranked values
- Format only values that are above or below average
- Format only unique or duplicate values
- Use a formula to determine which cells to format

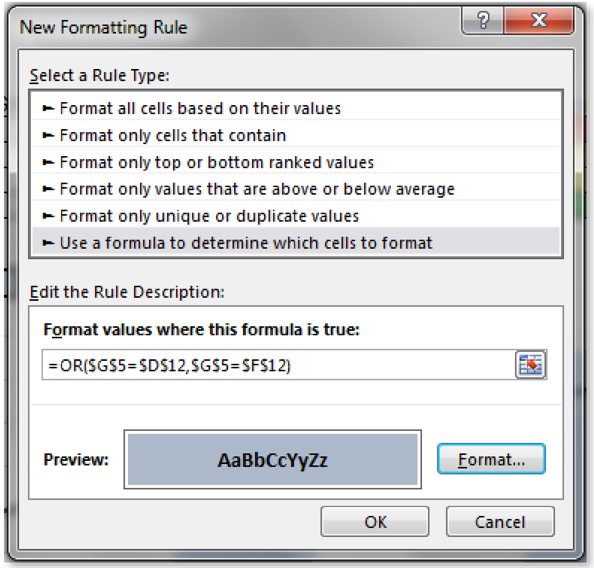
Edit the Rule Description:

Format all cells based on their values:

Format Style: 2-Color Scale

	Minimum	Maximum
Type:	Lowest Value	Highest Value
Value:	(Lowest value)	(Highest value)
Color:	🟧	🟨

Preview:



Select a Rule Type:

- Format all cells based on their values
- Format only cells that contain
- Format only top or bottom ranked values
- Format only values that are above or below average
- Format only unique or duplicate values
- Use a formula to determine which cells to format

Edit the Rule Description:

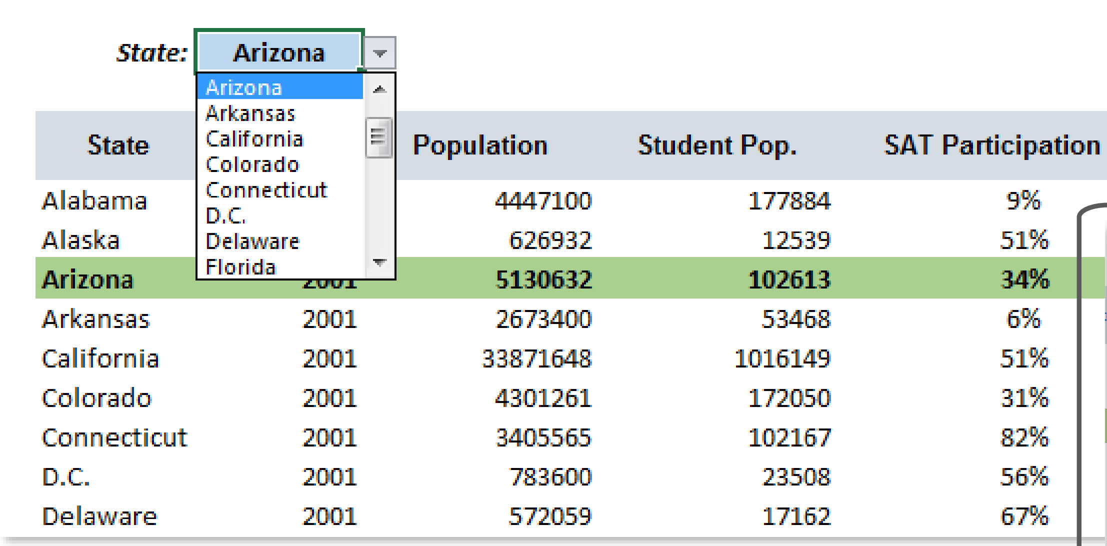
Format values where this formula is true:

=OR($G$5=$D$12,$G$5=$F$12)

Preview: AaBbCcYyZz

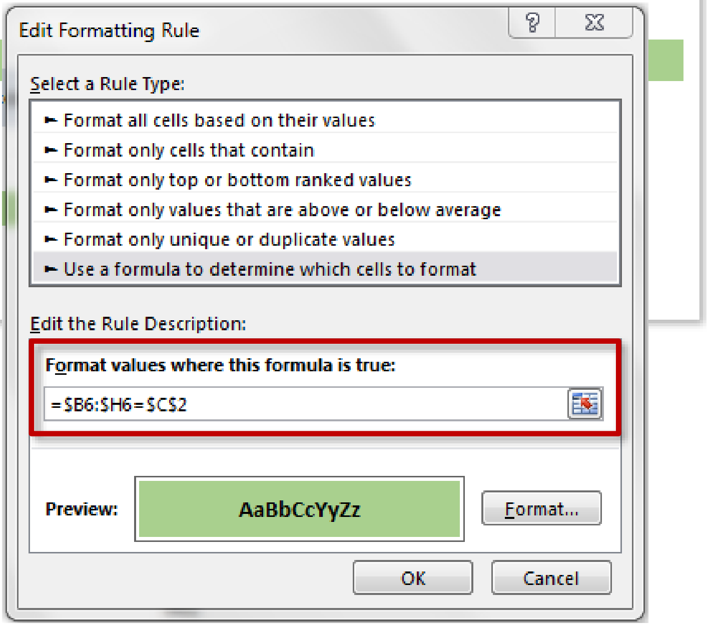
This is where you can add, clear, and manage your conditional formatting rules

Formula-Based Formatting



State: Arizona

State		Population	Student Pop.	SAT Participation Rate	Mean Verbal Score	Mean Math Score
Alabama		4447100	177884	9%	559	554
Alaska		626932	12539	51%		
Arizona	**2001**	**5130632**	**102613**	**34%**		
Arkansas	2001	2673400	53468	6%		
California	2001	33871648	1016149	51%		
Colorado	2001	4301261	172050	31%		
Connecticut	2001	3405565	102167	82%		
D.C.	2001	783600	23508	56%		
Delaware	2001	572059	17162	67%		



Edit Formatting Rule

Select a Rule Type:

- Format all cells based on their values
- Format only cells that contain
- Format only top or bottom ranked values
- Format only values that are above or below average
- Format only unique or duplicate values
- Use a formula to determine which cells to format

Edit the Rule Description:

Format values where this formula is true:

=$B6:$H6=$C$2

Preview: AaBbCcYyZz

In this example we're formatting the cells in columns B through H with a green fill and bold text, but only when the state name is equal to the value in cell $C$2

Note that the row label is relative (no "$"), which allows us to apply this formatting to other rows without losing functionality

MAVEN ANALYTICS

DYNAMIC ARRAY FORMULAS

DYNAMIC ARRAYS

In this section we'll introduce **dynamic arrays**, explore how array calculations work, and apply powerful functions like FILTER, SORT, and UNIQUE to explore and analyze data in Excel

TOPICS WE'LL COVER:

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

GOALS FOR THIS SECTION:

- *Understand how dynamic array calculations and spill ranges work*
- *Apply powerful array functions like FILTER, SORT, SEQUENCE and UNIQUE*
- *Learn how dynamic calculations can be applied to "legacy" functions in Excel*
- *Analyze data by combining dynamic array functions*

VERSIONS & COMPATIBILITY

✓ Dynamic array (DA) formulas are only available for **Office 365 subscribers**

- Users with standalone versions of Excel will not be able to use them

✓ Traditional CTRL+SHIFT+ENTER (CSE) array formulas are **no longer needed**

- You only need to press Enter for dynamic arrays (like any other function)
- For compatibility purposes, traditional CSE formulas will still work

✓ DA functions & spilled range references **won’t work in older Excel versions**

- Backwards compatibility is limited, and can lead to workbook errors
- Be mindful of this when creating workbooks that are meant to be shared!

PRO TIP: To verify whether or not you have access to dynamic array formulas, type a new DA function (like **=FILTER** or **=UNIQUE**) into a worksheet cell to see if Excel recognizes it

THE SECTION PROJECT

THE **SITUATION**

You’ve just been hired as a Business Intelligence Analyst for **Maven Recruiters***, a job placement agency based in the United States.

THE **BRIEF**

Your first task is to analyze a public dataset from **Glassdoor**, which tracks average salaries and growth rates across a range of industries and job titles.

Your goal is to use **dynamic array formulas** in Excel to explore the job landscape, compare salary expectations, and identify high-growth opportunities for the agency.

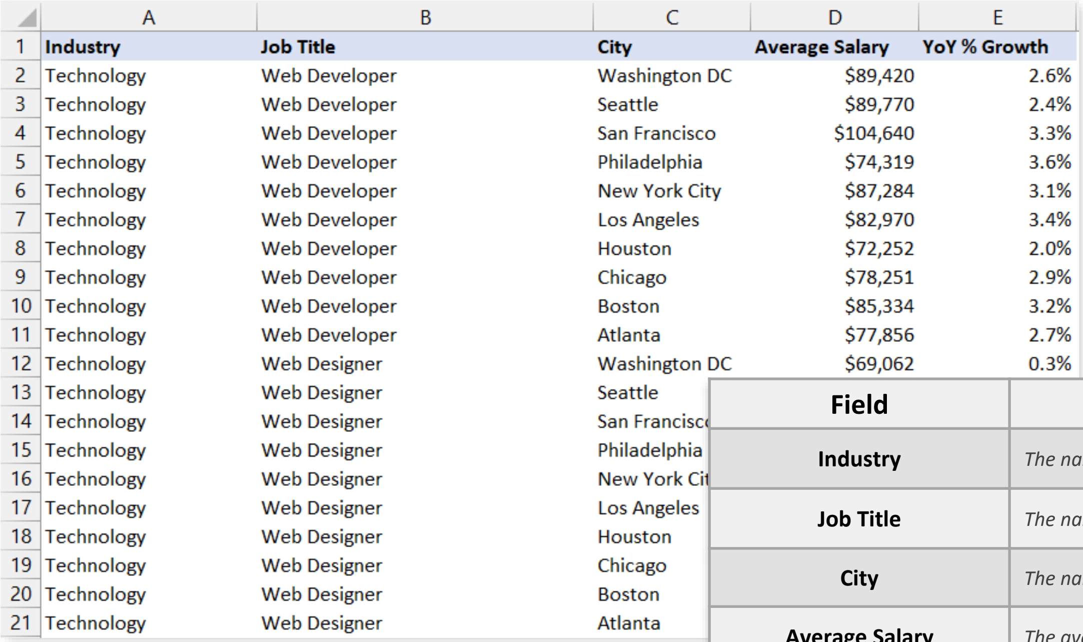
THE **OBJECTIVES**

- Identify top industries by salary and growth rate
- Review popular job titles by industry
- Create a tool to explore random samples of job titles
- Visualize the overall salary distribution
- Compare average job salaries across top US cities

glassdoor

*This data is adapted from Super Data Science sample data and is for informational purposes only . These samples are provided “as is” without warranty of any kind.

THE GLASSDOOR DATASET



	A	B	C	D	E
1	**Industry**	**Job Title**	**City**	**Average Salary**	**YoY % Growth**
2	Technology	Web Developer	Washington DC	$89,420	2.6%
3	Technology	Web Developer	Seattle	$89,770	2.4%
4	Technology	Web Developer	San Francisco	$104,640	3.3%
5	Technology	Web Developer	Philadelphia	$74,319	3.6%
6	Technology	Web Developer	New York City	$87,284	3.1%
7	Technology	Web Developer	Los Angeles	$82,970	3.4%
8	Technology	Web Developer	Houston	$72,252	2.0%
9	Technology	Web Developer	Chicago	$78,251	2.9%
10	Technology	Web Developer	Boston	$85,334	3.2%
11	Technology	Web Developer	Atlanta	$77,856	2.7%
12	Technology	Web Designer	Washington DC	$69,062	0.3%
13	Technology	Web Designer	Seattle		
14	Technology	Web Designer	San Francisc		
15	Technology	Web Designer	Philadelphia		
16	Technology	Web Designer	New York Cit		
17	Technology	Web Designer	Los Angeles		
18	Technology	Web Designer	Houston		
19	Technology	Web Designer	Chicago		
20	Technology	Web Designer	Boston		
21	Technology	Web Designer	Atlanta		

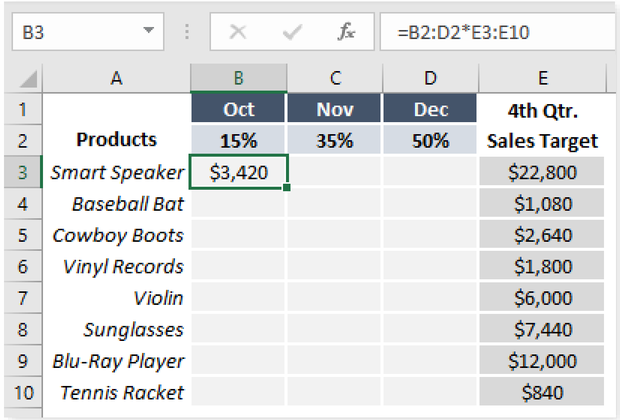
Field	Description
Industry	*The name of the industry, or job category, for each job title (17 total)*
Job Title	*The name of the job title (84 total)*
City	*The name of US cities where these job titles can be found (10 total)*
Average Salary	*The average salary in 2020 for each job title in each city*
YoY % Growth	*The percentage change in average salary from 2019 to 2020 for each job title in each city*

DYNAMIC EXCEL

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- RANDARRAY
- Legacy Functions

Excel's calculation engine has changed, and **formulas will never be the same**

BEFORE:
ONE formula = **ONE** value

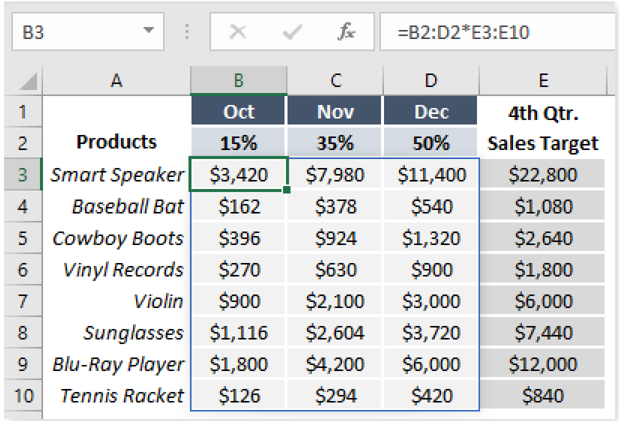


B3 =B2:D2*E3:E10

	A	B	C	D	E
1		Oct	Nov	Dec	4th Qtr.
2	Products	15%	35%	50%	Sales Target
3	*Smart Speaker*	$3,420			$22,800
4	*Baseball Bat*				$1,080
5	*Cowboy Boots*				$2,640
6	*Vinyl Records*				$1,800
7	*Violin*				$6,000
8	*Sunglasses*				$7,440
9	*Blu-Ray Player*				$12,000
10	*Tennis Racket*				$840

"Legacy" Excel

NOW:
ONE formula = **MANY** values



B3 =B2:D2*E3:E10

	A	B	C	D	E
1		Oct	Nov	Dec	4th Qtr.
2	Products	15%	35%	50%	Sales Target
3	*Smart Speaker*	$3,420	$7,980	$11,400	$22,800
4	*Baseball Bat*	$162	$378	$540	$1,080
5	*Cowboy Boots*	$396	$924	$1,320	$2,640
6	*Vinyl Records*	$270	$630	$900	$1,800
7	*Violin*	$900	$2,100	$3,000	$6,000
8	*Sunglasses*	$1,116	$2,604	$3,720	$7,440
9	*Blu-Ray Player*	$1,800	$4,200	$6,000	$12,000
10	*Tennis Racket*	$126	$294	$420	$840

"Dynamic" Excel

DYNAMIC EXCEL

Dynamic Excel
Spill Ranges
SORT & SORTBY
FILTER
UNIQUE
SEQUENCE
RANDARRAY
Legacy Functions

A single formula in a single cell can return ***many*** results

- Values “spill” across adjacent cells
- Results are not hard-coded, and can change dynamically with the source data

New **dynamic array functions** are now available

- Functions like SORT, FILTER and UNIQUE take advantage of the new calculation engine
- Combining dynamic array functions can unlock brand new capabilities in Excel

Dynamic array behavior applies to **traditional Excel formulas** as well

- Existing functions can also leverage Excel’s new calculation engine

Array functions are now **simpler and easier to learn**

- Traditional CSE array functions were typically only used by “experts”
- New dynamic array formulas are intuitive and user-friendly

DYNAMIC EXCEL

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

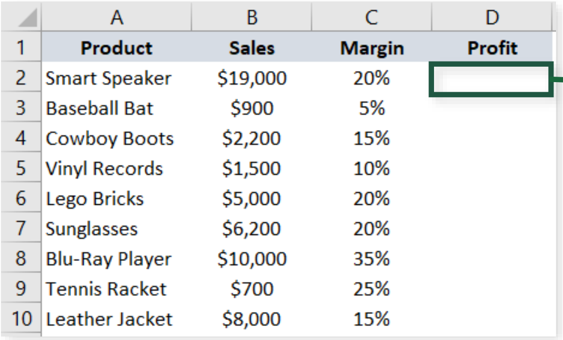
UNIQUE

SEQUENCE

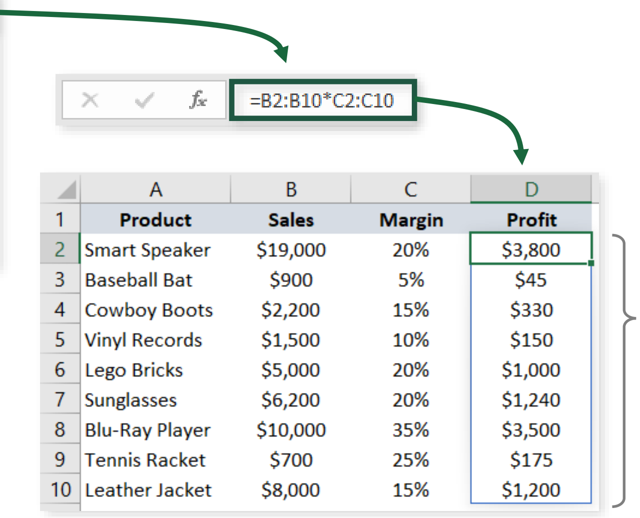
RANDARRAY

Legacy Functions

Formulas that can return arrays of variable size are called **dynamic arrays**; these formulas are entered in a single cell and can “spill” results across an entire range



	A	B	C	D
1	**Product**	**Sales**	**Margin**	**Profit**
2	Smart Speaker	$19,000	20%	
3	Baseball Bat	$900	5%	
4	Cowboy Boots	$2,200	15%	
5	Vinyl Records	$1,500	10%	
6	Lego Bricks	$5,000	20%	
7	Sunglasses	$6,200	20%	
8	Blu-Ray Player	$10,000	35%	
9	Tennis Racket	$700	25%	
10	Leather Jacket	$8,000	15%	



=B2:B10*C2:C10

	A	B	C	D
1	**Product**	**Sales**	**Margin**	**Profit**
2	Smart Speaker	$19,000	20%	$3,800
3	Baseball Bat	$900	5%	$45
4	Cowboy Boots	$2,200	15%	$330
5	Vinyl Records	$1,500	10%	$150
6	Lego Bricks	$5,000	20%	$1,000
7	Sunglasses	$6,200	20%	$1,240
8	Blu-Ray Player	$10,000	35%	$3,500
9	Tennis Racket	$700	25%	$175
10	Leather Jacket	$8,000	15%	$1,200

The resulting range of cells is known as the ***spill range***

HEY THIS IS IMPORTANT!

The formula itself only lives in the **first cell** of the spilled range

SPILL RANGE PROPERTIES

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

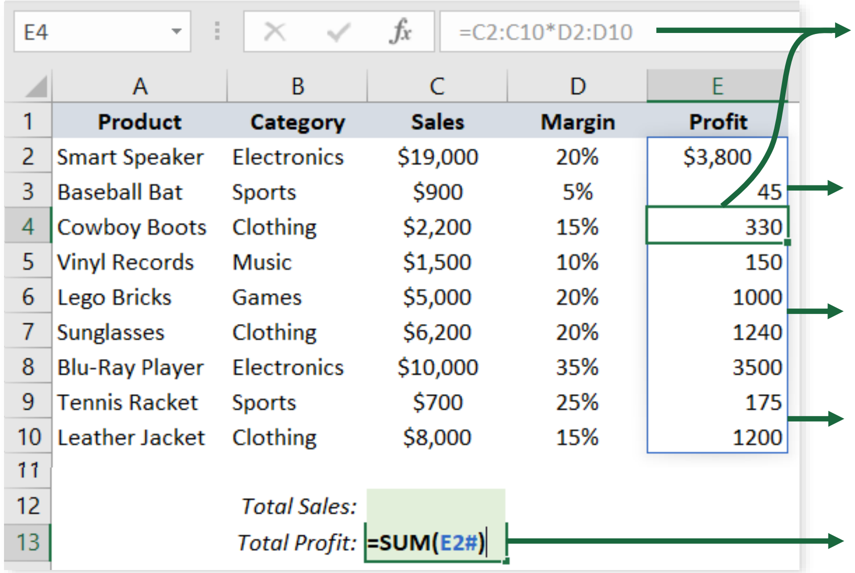
UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

The **spill range** contains the results of a single dynamic array formula



E4 =C2:C10*D2:D10

	A	B	C	D	E
1	**Product**	**Category**	**Sales**	**Margin**	**Profit**
2	Smart Speaker	Electronics	$19,000	20%	$3,800
3	Baseball Bat	Sports	$900	5%	45
4	Cowboy Boots	Clothing	$2,200	15%	330
5	Vinyl Records	Music	$1,500	10%	150
6	Lego Bricks	Games	$5,000	20%	1000
7	Sunglasses	Clothing	$6,200	20%	1240
8	Blu-Ray Player	Electronics	$10,000	35%	3500
9	Tennis Racket	Sports	$700	25%	175
10	Leather Jacket	Clothing	$8,000	15%	1200
11					
12		*Total Sales:*			
13		*Total Profit:*	=SUM(E2#)		

Only the **first cell** in the range is **editable** *(all others within the spill range are grayed out)*

Cell formatting isn't carried over from the source and **does not spill**

Spill ranges are highlighted with a **blue border** when selected

Spilled ranges **update automatically** and **resize** to fit the resulting array

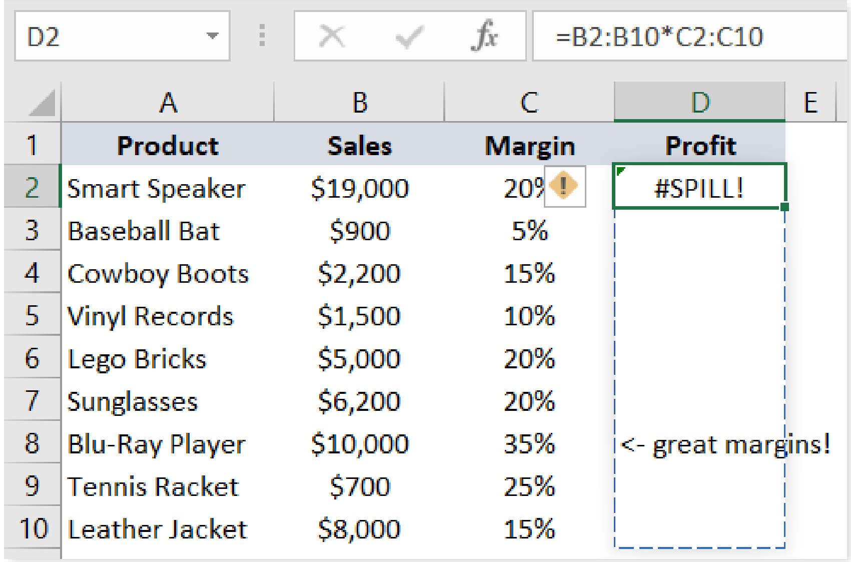
The "**#**" symbol can be used to **reference an entire spilled range**

#SPILL! ERRORS

- Dynamic Excel
- **Spill Ranges**
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- RANDARRAY
- Legacy Functions

#SPILL! errors occur when something is “blocking” the spill range, since Excel will not overwrite existing values by default

- Clear any existing text, values, or merged cells from the entire spill range to fix the error



D2 | *fx* =B2:B10*C2:C10

	A	B	C	D	E
1	Product	Sales	Margin	Profit	
2	Smart Speaker	$19,000	20%	#SPILL!	
3	Baseball Bat	$900	5%		
4	Cowboy Boots	$2,200	15%		
5	Vinyl Records	$1,500	10%		
6	Lego Bricks	$5,000	20%		
7	Sunglasses	$6,200	20%		
8	Blu-Ray Player	$10,000	35%	<- great margins!	
9	Tennis Racket	$700	25%		
10	Leather Jacket	$8,000	15%		

HEY THIS IS IMPORTANT!

You may also see a #SPILL! error if you try to use dynamic array formulas **inside of a table**

MAVEN ANALYTICS

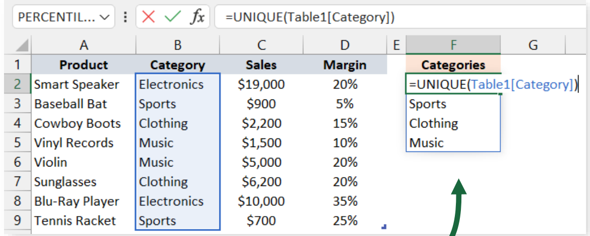
PRO TIP: GROWING SOURCE DATA

References within dynamic array formulas **do not automatically resize** as you add new data, but there are several ways to accommodate dynamic source data:

1. Format the data as a **table** and use structured references

2. Select the **extra rows** to accommodate new records

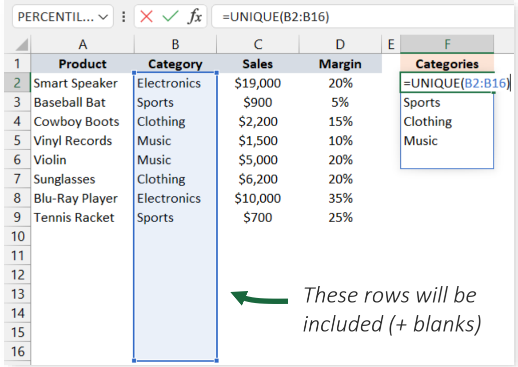
3. Define a dynamic **named range** using OFFSET & COUNTA



PERCENTIL... =UNIQUE(Table1[Category])

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**	**Margin**		**Categories**	
2	Smart Speaker	Electronics	$19,000	20%		=UNIQUE(Table1[Category])	
3	Baseball Bat	Sports	$900	5%		Sports	
4	Cowboy Boots	Clothing	$2,200	15%		Clothing	
5	Vinyl Records	Music	$1,500	10%		Music	
6	Violin	Music	$5,000	20%			
7	Sunglasses	Clothing	$6,200	20%			
8	Blu-Ray Player	Electronics	$10,000	35%			
9	Tennis Racket	Sports	$700	25%			

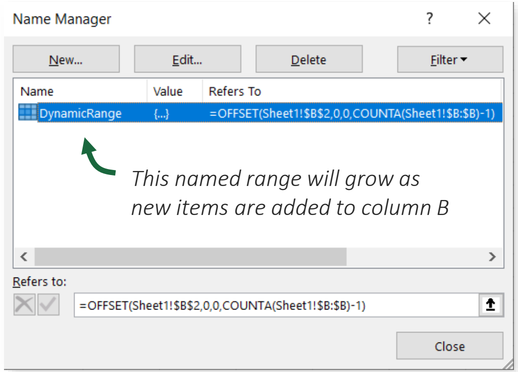
Any new rows will be included



PERCENTIL... =UNIQUE(B2:B16)

	A	B	C	D	E	F
1	**Product**	**Category**	**Sales**	**Margin**		**Categories**
2	Smart Speaker	Electronics	$19,000	20%		=UNIQUE(B2:B16)
3	Baseball Bat	Sports	$900	5%		Sports
4	Cowboy Boots	Clothing	$2,200	15%		Clothing
5	Vinyl Records	Music	$1,500	10%		Music
6	Violin	Music	$5,000	20%		
7	Sunglasses	Clothing	$6,200	20%		
8	Blu-Ray Player	Electronics	$10,000	35%		
9	Tennis Racket	Sports	$700	25%		
10						
11						
12						
13						
14						
15						
16						

These rows will be included (+ blanks)



Name	Value	Refers To
DynamicRange	{...}	=OFFSET(Sheet1!$B$2,0,0,COUNTA(Sheet1!$B:$B)-1)

Refers to: =OFFSET(Sheet1!$B$2,0,0,COUNTA(Sheet1!$B:$B)-1)

This named range will grow as new items are added to column B

DYNAMIC ARRAY FUNCTIONS

SORT()

Sorts an array of data by one or more columns in the array

SORTBY()

Sorts an array of data by one or more columns in another array

FILTER()

Filters an array of data based on specified criteria and returns the matching records

UNIQUE()

Removes duplicates from an array of data and returns the unique records

SEQUENCE()

Generates a one- or two-dimensional array of sequential numbers

RANDARRAY()

Generates a one- or two-dimensional array of random numbers

SORT

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

SORT()

Sorts an array of data by one or more columns in the array

=**SORT** (array, *[sort_index]*, *[sort_order]*, *[by_col]*)

An array of cells that you want to sort

Column # you want to sort by

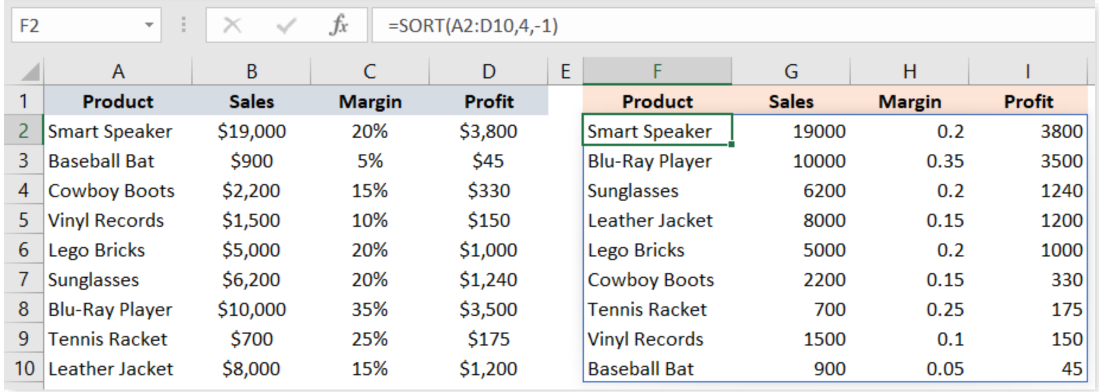
(Default is 1)

1 *= Ascending*
-1 *= Descending*

(Default is 1)

TRUE/1 *= Sort by column*
FALSE/0 *= Sort by row*

(Default is FALSE or 0)



F2 =SORT(A2:D10,4,-1)

	A	B	C	D	E	F	G	H	I
1	Product	Sales	Margin	Profit		Product	Sales	Margin	Profit
2	Smart Speaker	$19,000	20%	$3,800		Smart Speaker	19000	0.2	3800
3	Baseball Bat	$900	5%	$45		Blu-Ray Player	10000	0.35	3500
4	Cowboy Boots	$2,200	15%	$330		Sunglasses	6200	0.2	1240
5	Vinyl Records	$1,500	10%	$150		Leather Jacket	8000	0.15	1200
6	Lego Bricks	$5,000	20%	$1,000		Lego Bricks	5000	0.2	1000
7	Sunglasses	$6,200	20%	$1,240		Cowboy Boots	2200	0.15	330
8	Blu-Ray Player	$10,000	35%	$3,500		Tennis Racket	700	0.25	175
9	Tennis Racket	$700	25%	$175		Vinyl Records	1500	0.1	150
10	Leather Jacket	$8,000	15%	$1,200		Baseball Bat	900	0.05	45

The array in ***A2:D10*** *is being sorted by the* ***4th column*** *(Profit) in* ***descending*** *order*

SORT

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- RANDARRAY
- Legacy Functions

SORT()

Sorts an array of data by one or more columns in the array

=**SORT** (array, *[sort_index]*, *[sort_order]*, *[by_col]*)

An array of cells that you want to sort

Column # you want to sort by

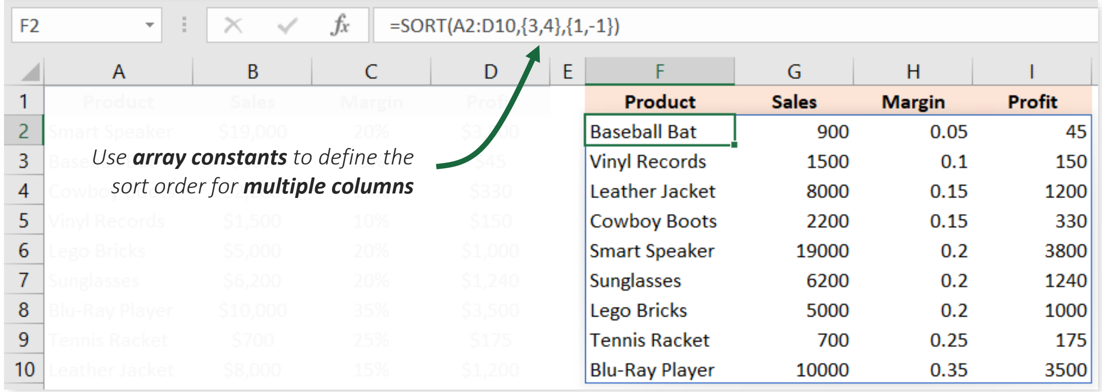
(Default is 1)

1 *= Ascending*
-1 *= Descending*

(Default is 1)

TRUE/1 *= Sort by column*
FALSE/0 *= Sort by row*

(Default is FALSE or 0)



F2

=SORT(A2:D10,{3,4},{1,-1})

	A	B	C	D	E	F	G	H	I
1	Product	Sales	Margin	Profit		**Product**	**Sales**	**Margin**	**Profit**
2	Smart Speaker	$19,000	20%	$3,800		Baseball Bat	900	0.05	45
3	Baseball Bat	$900	5%	$45		Vinyl Records	1500	0.1	150
4	Cowboy Boots	$2,200	15%	$330		Leather Jacket	8000	0.15	1200
5	Vinyl Records	$1,500	10%	$150		Cowboy Boots	2200	0.15	330
6	Lego Bricks	$5,000	20%	$1,000		Smart Speaker	19000	0.2	3800
7	Sunglasses	$6,200	20%	$1,240		Sunglasses	6200	0.2	1240
8	Blu-Ray Player	$10,000	35%	$3,500		Lego Bricks	5000	0.2	1000
9	Tennis Racket	$700	25%	$175		Tennis Racket	700	0.25	175
10	Leather Jacket	$8,000	15%	$1,200		Blu-Ray Player	10000	0.35	3500

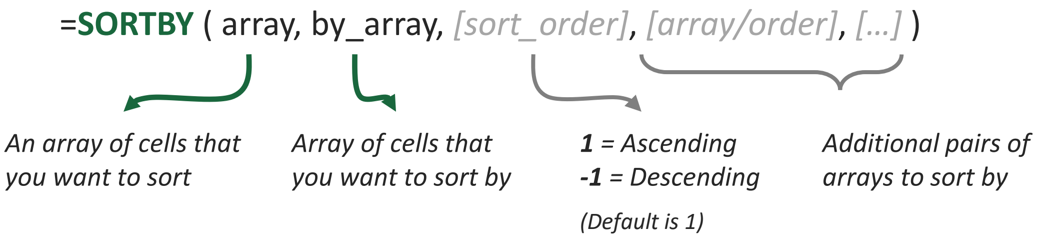
Use ***array constants*** *to define the sort order for* ***multiple columns***

The array in **A2:D10** *is being sorted by the 3rd* **column** *(Margin) in* ***ascending*** *order, then the 4th* **column** *(Profit) in* ***descending*** *order*

SORTBY

SORTBY()

Sorts an array of data by one or more columns in another array



=**SORTBY** (array, by_array, *[sort_order]*, *[array/order]*, *[…]*)

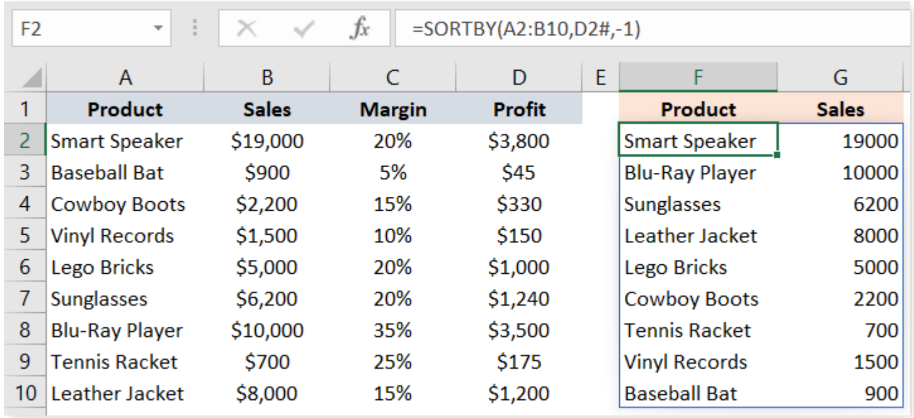
An array of cells that you want to sort

Array of cells that you want to sort by

***1** = Ascending*
***-1** = Descending*

(Default is 1)

Additional pairs of arrays to sort by



F2

=SORTBY(A2:B10,D2#,-1)

	A	B	C	D	E	F	G
1	**Product**	**Sales**	**Margin**	**Profit**		**Product**	**Sales**
2	Smart Speaker	$19,000	20%	$3,800		Smart Speaker	19000
3	Baseball Bat	$900	5%	$45		Blu-Ray Player	10000
4	Cowboy Boots	$2,200	15%	$330		Sunglasses	6200
5	Vinyl Records	$1,500	10%	$150		Leather Jacket	8000
6	Lego Bricks	$5,000	20%	$1,000		Lego Bricks	5000
7	Sunglasses	$6,200	20%	$1,240		Cowboy Boots	2200
8	Blu-Ray Player	$10,000	35%	$3,500		Tennis Racket	700
9	Tennis Racket	$700	25%	$175		Vinyl Records	1500
10	Leather Jacket	$8,000	15%	$1,200		Baseball Bat	900

*This array is sorted by **Profit** in **descending** order (even though it isn't in the array!)*

HEY THIS IS IMPORTANT!

The array you sort by must be the **same size** as the array you are sorting

FILTER

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- **FILTER**
- UNIQUE
- SEQUENCE
- RANDARRAY
- Legacy Functions

FILTER()

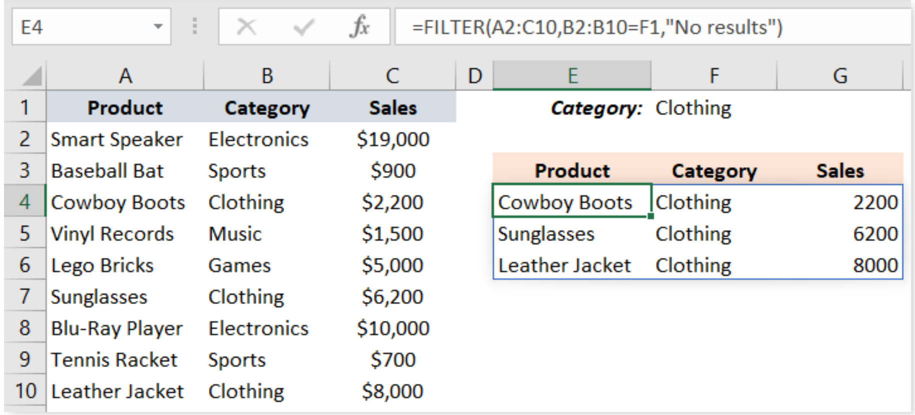
Filters an array of data based on specified criteria and returns the matching records

=**FILTER** (array, include, *[if_empty]*)

An array of cells that you want to filter

*A logical test to determine the filter criteria, where values of **TRUE** will be kept*

An optional value to return if nothing passes the filter criteria



E4 =FILTER(A2:C10,B2:B10=F1,"No results")

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**		***Category:***	Clothing	
2	Smart Speaker	Electronics	$19,000				
3	Baseball Bat	Sports	$900		**Product**	**Category**	**Sales**
4	Cowboy Boots	Clothing	$2,200		Cowboy Boots	Clothing	2200
5	Vinyl Records	Music	$1,500		Sunglasses	Clothing	6200
6	Lego Bricks	Games	$5,000		Leather Jacket	Clothing	8000
7	Sunglasses	Clothing	$6,200				
8	Blu-Ray Player	Electronics	$10,000				
9	Tennis Racket	Sports	$700				
10	Leather Jacket	Clothing	$8,000				

*This array returns values from **A2:C10**, where Category = **Clothing***

FILTER

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

FILTER()

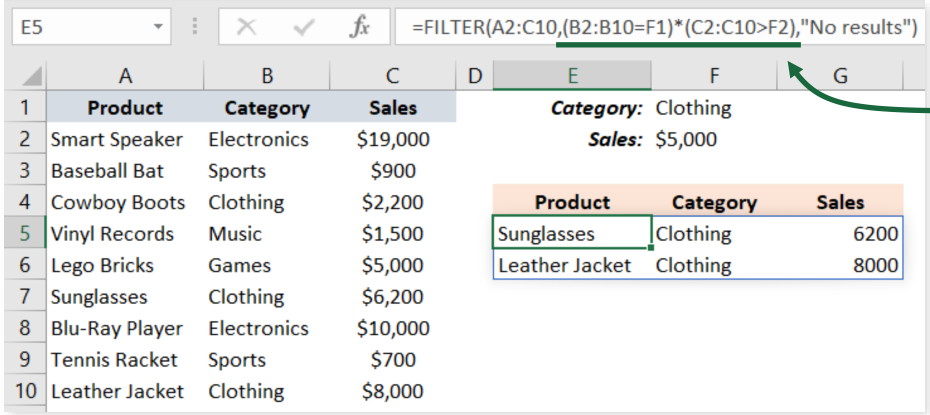
Filters an array of data based on specified criteria and returns the matching records

=**FILTER** (array, include, *[if_empty]*)

An array of cells that you want to filter

*A logical test to determine the filter criteria, where values of **TRUE** will be kept*

An optional value to return if nothing passes the filter criteria



E5 =FILTER(A2:C10,(B2:B10=F1)*(C2:C10>F2),"No results")

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**		***Category:***	Clothing	
2	Smart Speaker	Electronics	$19,000		***Sales:***	$5,000	
3	Baseball Bat	Sports	$900				
4	Cowboy Boots	Clothing	$2,200		**Product**	**Category**	**Sales**
5	Vinyl Records	Music	$1,500		Sunglasses	Clothing	6200
6	Lego Bricks	Games	$5,000		Leather Jacket	Clothing	8000
7	Sunglasses	Clothing	$6,200				
8	Blu-Ray Player	Electronics	$10,000				
9	Tennis Racket	Sports	$700				
10	Leather Jacket	Clothing	$8,000				

*To create an **AND** condition between multiple logical tests, you can **multiply** them together*

*This array returns values from **A2:C10** where Category = **Clothing** AND Sales **> 5,000***

(BOTH criteria must be met)

FILTER

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- RANDARRAY
- Legacy Functions

FILTER()

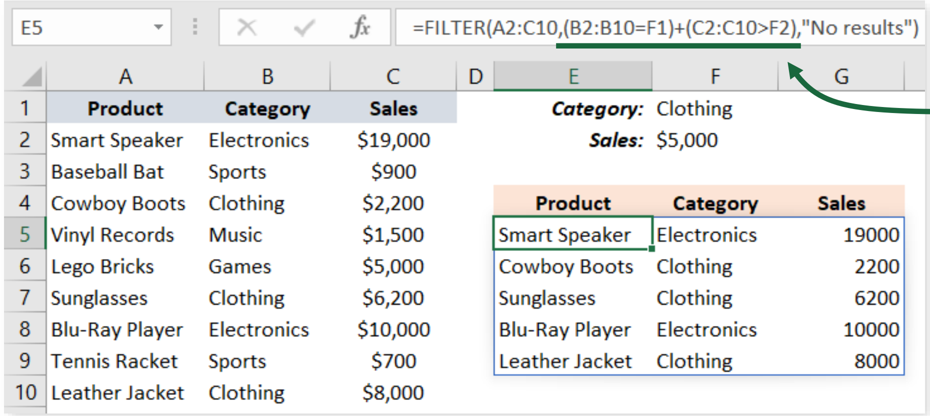
Filters an array of data based on specified criteria and returns the matching records

=**FILTER** (array, include, *[if_empty]*)

An array of cells that you want to filter

*A logical test to determine the filter criteria, where values of **TRUE** will be kept*

An optional value to return if nothing passes the filter criteria



E5

=FILTER(A2:C10,(B2:B10=F1)+(C2:C10>F2),"No results")

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**		***Category:***	Clothing	
2	Smart Speaker	Electronics	$19,000		***Sales:***	$5,000	
3	Baseball Bat	Sports	$900				
4	Cowboy Boots	Clothing	$2,200		**Product**	**Category**	**Sales**
5	Vinyl Records	Music	$1,500		Smart Speaker	Electronics	19000
6	Lego Bricks	Games	$5,000		Cowboy Boots	Clothing	2200
7	Sunglasses	Clothing	$6,200		Sunglasses	Clothing	6200
8	Blu-Ray Player	Electronics	$10,000		Blu-Ray Player	Electronics	10000
9	Tennis Racket	Sports	$700		Leather Jacket	Clothing	8000
10	Leather Jacket	Clothing	$8,000				

*To create an **OR** condition between multiple logical tests, you can **sum** them together*

*This array returns values from **A2:C10** where Category = **Clothing** OR Sales **> 5,000***

(EITHER criteria must be met)

MAVEN ANALYTICS

UNIQUE

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

UNIQUE()

Removes duplicates from an array of data and returns only the unique records

=**UNIQUE** (array, *[by_col]*, *[exactly_once]*)

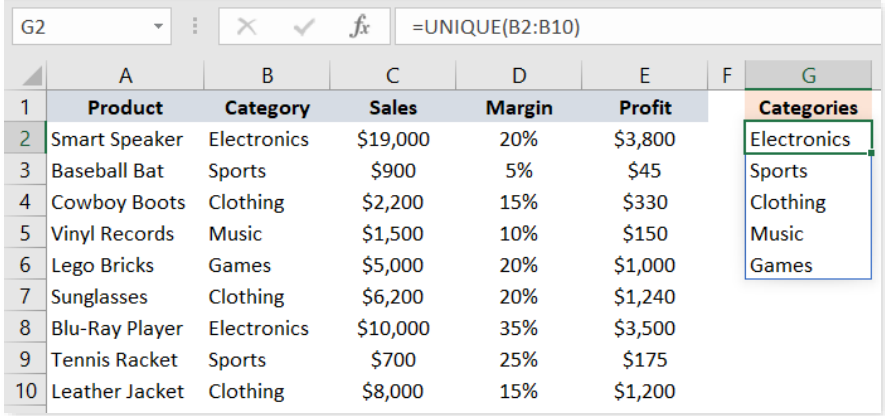
An array of cells that you want to remove duplicates from

TRUE/1 *= Remove duplicates in columns*
FALSE/0 *= Remove duplicates in rows*

(Default is FALSE or 0)

TRUE/1 *= Extract values that* ***only appear once***
FALSE/0 *= Extract all unique values*

(Default is FALSE or 0)



G2 =UNIQUE(B2:B10)

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**	**Margin**	**Profit**		**Categories**
2	Smart Speaker	Electronics	$19,000	20%	$3,800		Electronics
3	Baseball Bat	Sports	$900	5%	$45		Sports
4	Cowboy Boots	Clothing	$2,200	15%	$330		Clothing
5	Vinyl Records	Music	$1,500	10%	$150		Music
6	Lego Bricks	Games	$5,000	20%	$1,000		Games
7	Sunglasses	Clothing	$6,200	20%	$1,240		
8	Blu-Ray Player	Electronics	$10,000	35%	$3,500		
9	Tennis Racket	Sports	$700	25%	$175		
10	Leather Jacket	Clothing	$8,000	15%	$1,200		

This array returns the unique Category values from ***B2:B10***

UNIQUE

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- **UNIQUE**
- SEQUENCE
- RANDARRAY
- Legacy Functions

UNIQUE()

Removes duplicates from an array of data and returns only the unique records

=**UNIQUE** (array, *[by_col]*, *[exactly_once]*)

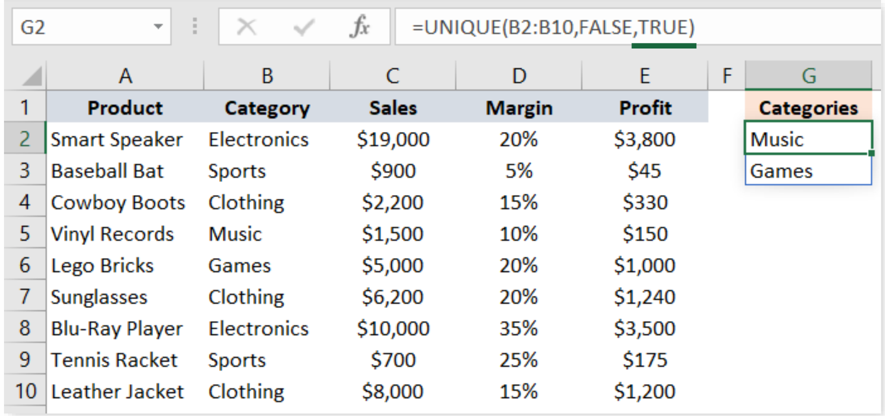
An array of cells that you want to remove duplicates from

TRUE/1 *= Remove duplicates in columns*
FALSE/0 *= Remove duplicates in rows*

(Default is FALSE or 0)

TRUE/1 *= Extract values that* ***only appear once***
FALSE/0 *= Extract all unique values*

(Default is FALSE or 0)



G2 | =UNIQUE(B2:B10,FALSE,TRUE)

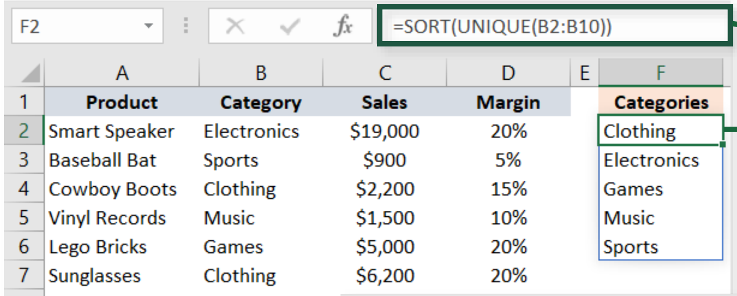
	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**	**Margin**	**Profit**		**Categories**
2	Smart Speaker	Electronics	$19,000	20%	$3,800		Music
3	Baseball Bat	Sports	$900	5%	$45		Games
4	Cowboy Boots	Clothing	$2,200	15%	$330		
5	Vinyl Records	Music	$1,500	10%	$150		
6	Lego Bricks	Games	$5,000	20%	$1,000		
7	Sunglasses	Clothing	$6,200	20%	$1,240		
8	Blu-Ray Player	Electronics	$10,000	35%	$3,500		
9	Tennis Racket	Sports	$700	25%	$175		
10	Leather Jacket	Clothing	$8,000	15%	$1,200		

This array returns the Category values from ***B2:B10*** *that* ***appear exactly once***

PRO TIP: Include multiple columns in the array to return each unique **combination** of values

PRO TIP: Combining SORT, FILTER & UNIQUE

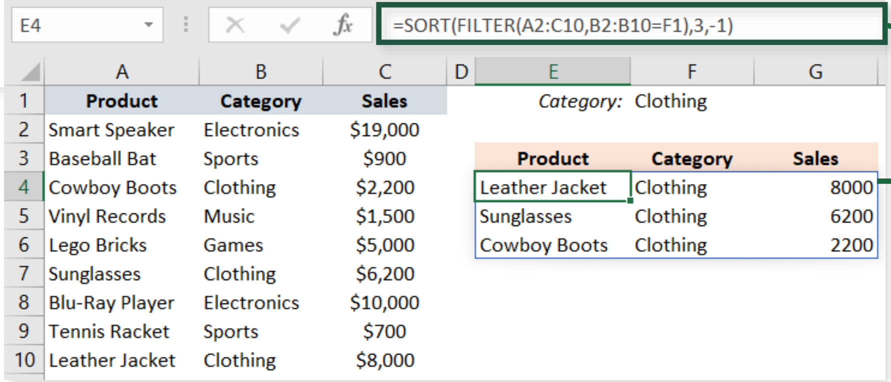
You can **combine**, or “nest”, multiple dynamic array functions to perform multiple operations



F2 =SORT(UNIQUE(B2:B10))

	A	B	C	D	E	F
1	**Product**	**Category**	**Sales**	**Margin**		**Categories**
2	Smart Speaker	Electronics	$19,000	20%		Clothing
3	Baseball Bat	Sports	$900	5%		Electronics
4	Cowboy Boots	Clothing	$2,200	15%		Games
5	Vinyl Records	Music	$1,500	10%		Music
6	Lego Bricks	Games	$5,000	20%		Sports
7	Sunglasses	Clothing	$6,200	20%		
8	Blu-Ray Player	Electronics				
9	Tennis Racket	Sports				
10	Leather Jacket	Clothing				

*Here we’re combining **SORT** and **UNIQUE** to return an array of unique categories in ascending order*



E4 =SORT(FILTER(A2:C10,B2:B10=F1),3,-1)

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**		*Category:*	Clothing	
2	Smart Speaker	Electronics	$19,000				
3	Baseball Bat	Sports	$900		**Product**	**Category**	**Sales**
4	Cowboy Boots	Clothing	$2,200		Leather Jacket	Clothing	8000
5	Vinyl Records	Music	$1,500		Sunglasses	Clothing	6200
6	Lego Bricks	Games	$5,000		Cowboy Boots	Clothing	2200
7	Sunglasses	Clothing	$6,200				
8	Blu-Ray Player	Electronics	$10,000				
9	Tennis Racket	Sports	$700				
10	Leather Jacket	Clothing	$8,000				

*Here we’re combining **SORT** and **FILTER** to return an array of products in the Clothing category, sorted by Sales*

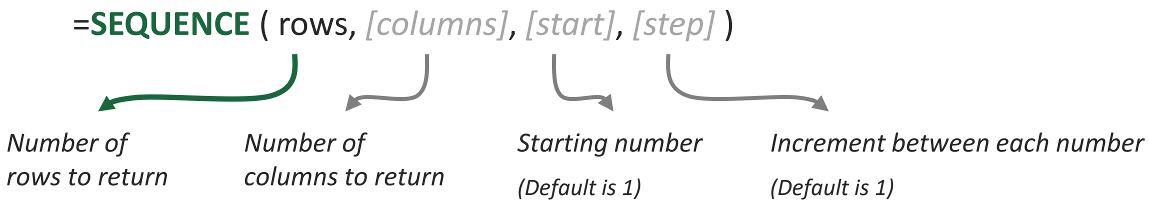
MAVEN ANALYTICS

SEQUENCE

Dynamic Excel
Spill Ranges
SORT & SORTBY
FILTER
UNIQUE
SEQUENCE
RANDARRAY
Legacy Functions

SEQUENCE()

Generates a one- or two-dimensional array of sequential numbers

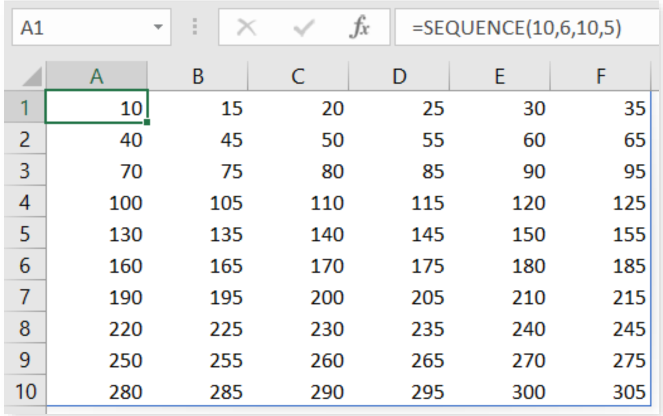

=**SEQUENCE** (rows, *[columns]*, *[start]*, *[step]*)

Number of rows to return

Number of columns to return

Starting number
(Default is 1)

Increment between each number
(Default is 1)



=SEQUENCE(10,6,10,5)

	A	B	C	D	E	F
1	10	15	20	25	30	35
2	40	45	50	55	60	65
3	70	75	80	85	90	95
4	100	105	110	115	120	125
5	130	135	140	145	150	155
6	160	165	170	175	180	185
7	190	195	200	205	210	215
8	220	225	230	235	240	245
9	250	255	260	265	270	275
10	280	285	290	295	300	305

*This generates a **10-row**, **6-column** array starting at **10** and incrementing by **5** (note the numbers go left-to-right, then down)*

PRO TIP: Nest **SEQUENCE** within other functions to make them more dynamic

RANDARRAY

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- **RANDARRAY**
- Legacy Functions

RANDARRAY()

Generates a one- or two-dimensional array of random numbers

=**RANDARRAY** (*[rows]*, *[columns]*, *[min]*, *[max]*, *[integer]*)

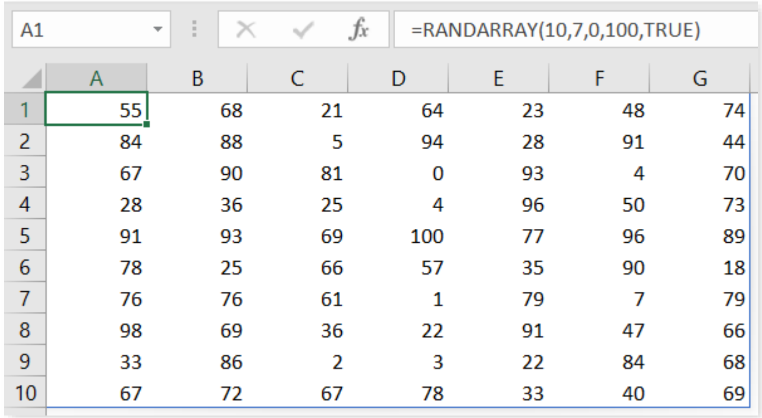
[rows]: *Number of rows to return* *(Default is 1)*

[columns]: *Number of columns to return* *(Default is 1)*

[min]: *Minimum value to return* *(Default is 0)*

[max]: *Maximum value to return* *(Default is 1)*

[integer]: *Return whole numbers?* *(Default is FALSE or 0)*



A1 | =RANDARRAY(10,7,0,100,TRUE)

	A	B	C	D	E	F	G
1	55	68	21	64	23	48	74
2	84	88	5	94	28	91	44
3	67	90	81	0	93	4	70
4	28	36	25	4	96	50	73
5	91	93	69	100	77	96	89
6	78	25	66	57	35	90	18
7	76	76	61	1	79	7	79
8	98	69	36	22	91	47	66
9	33	86	2	3	22	84	68
10	67	72	67	78	33	40	69

*This generates a **10-row** by **7-column** array of random **whole numbers** between **0** and **100***

PRO TIP: Use **RANDARRAY** to randomly sort lists of data

LEGACY FUNCTION: FREQUENCY

Dynamic Excel

Spill Ranges

SORT & SORTBY

FILTER

UNIQUE

SEQUENCE

RANDARRAY

Legacy Functions

FREQUENCY()

Returns the frequency of values in a range based on specified intervals (or bins)

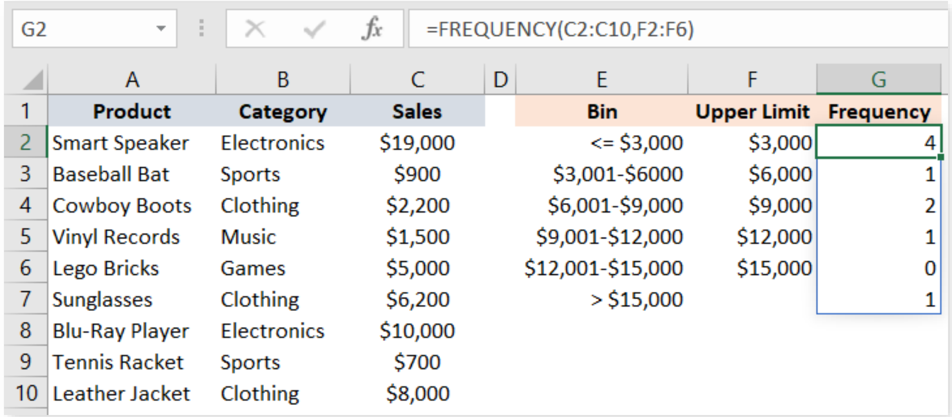
=**FREQUENCY** (data_array, bins_array)

An array of cells containing values

An array of intervals (bins) for grouping the values

HEY THIS IS IMPORTANT!

The **bins_array** argument must include the upper limit of each bin, formatted as a number



G2 =FREQUENCY(C2:C10,F2:F6)

	A	B	C	D	E	F	G
1	**Product**	**Category**	**Sales**		**Bin**	**Upper Limit**	**Frequency**
2	Smart Speaker	Electronics	$19,000		<= $3,000	$3,000	4
3	Baseball Bat	Sports	$900		$3,001-$6000	$6,000	1
4	Cowboy Boots	Clothing	$2,200		$6,001-$9,000	$9,000	2
5	Vinyl Records	Music	$1,500		$9,001-$12,000	$12,000	1
6	Lego Bricks	Games	$5,000		$12,001-$15,000	$15,000	0
7	Sunglasses	Clothing	$6,200		> $15,000		1
8	Blu-Ray Player	Electronics	$10,000				
9	Tennis Racket	Sports	$700				
10	Leather Jacket	Clothing	$8,000				

Here we're counting the frequency of Sales records which fall into each bin in ***F2:F6***

NOTE*: FREQUENCY always returns* ***one extra row*** *to account for values above the largest defined bin*

LEGACY FUNCTION: TRANSPOSE

- Dynamic Excel
- Spill Ranges
- SORT & SORTBY
- FILTER
- UNIQUE
- SEQUENCE
- RANDARRAY
- **Legacy Functions**

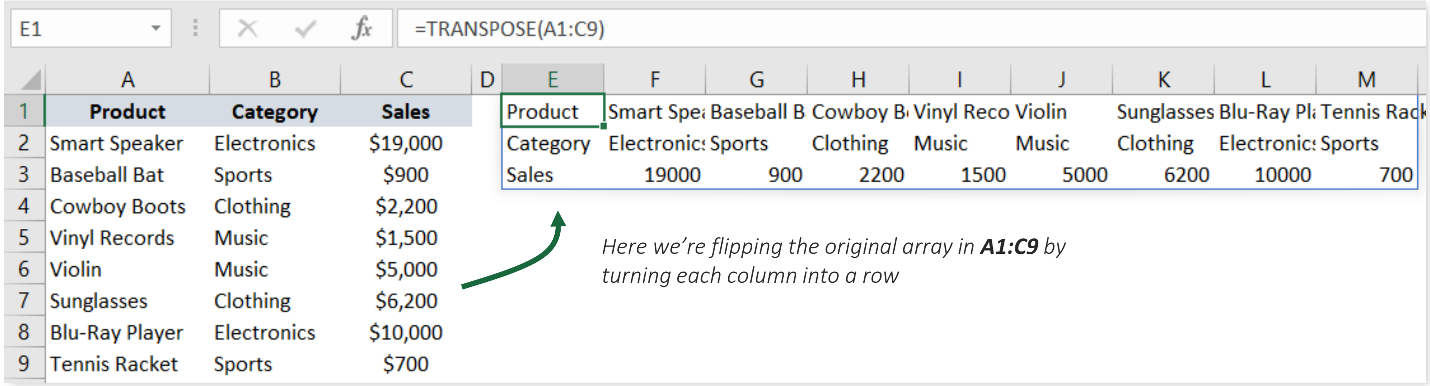
TRANSPOSE()

Flips a vertical range of cells to a horizontal range, or vice versa

=**TRANSPOSE** (array)

An array of cells you want to transpose

PRO TIP: Use **TRANSPOSE** to "pivot" your data by turning rows into columns, or vice versa



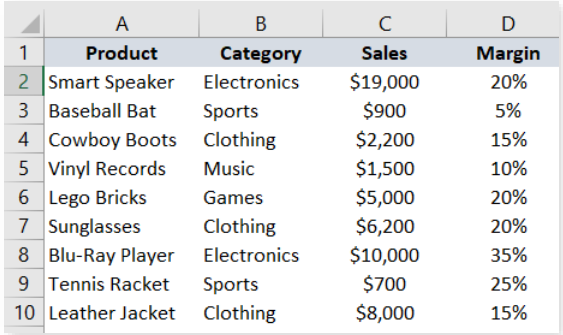
E1 =TRANSPOSE(A1:C9)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Product**	**Category**	**Sales**		Product	Smart Spe	Baseball B	Cowboy B	Vinyl Reco	Violin	Sunglasses	Blu-Ray Pl	Tennis Rack
2	Smart Speaker	Electronics	$19,000		Category	Electronic	Sports	Clothing	Music	Music	Clothing	Electronic	Sports
3	Baseball Bat	Sports	$900		Sales	19000	900	2200	1500	5000	6200	10000	700
4	Cowboy Boots	Clothing	$2,200										
5	Vinyl Records	Music	$1,500										
6	Violin	Music	$5,000										
7	Sunglasses	Clothing	$6,200										
8	Blu-Ray Player	Electronics	$10,000										
9	Tennis Racket	Sports	$700										

Here we're flipping the original array in ***A1:C9*** *by turning each column into a row*

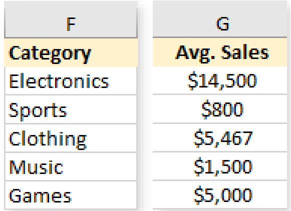
PRO TIP: JOINING ARRAYS WITH CHOOSE

The **CHOOSE** function can be used to combine separate cell ranges into a single array, which can be referenced or manipulated within other formulas



	A	B	C	D
1	**Product**	**Category**	**Sales**	**Margin**
2	Smart Speaker	Electronics	$19,000	20%
3	Baseball Bat	Sports	$900	5%
4	Cowboy Boots	Clothing	$2,200	15%
5	Vinyl Records	Music	$1,500	10%
6	Lego Bricks	Games	$5,000	20%
7	Sunglasses	Clothing	$6,200	20%
8	Blu-Ray Player	Electronics	$10,000	35%
9	Tennis Racket	Sports	$700	25%
10	Leather Jacket	Clothing	$8,000	15%

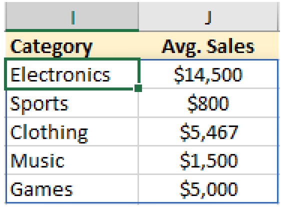
Raw data at the product-level



F	G
Category	**Avg. Sales**
Electronics	$14,500
Sports	$800
Clothing	$5,467
Music	$1,500
Games	$5,000

Avg Sales by Category (using ***AVERAGEIF****)*

Unique list of Categories (using ***UNIQUE****)*



I	J
Category	**Avg. Sales**
Electronics	$14,500
Sports	$800
Clothing	$5,467
Music	$1,500
Games	$5,000

=**CHOOSE**({1,2},F2#,G2#)

PRO TIP: Using the array constant **{1,2}** tells Excel to combine both referenced ranges into one dynamic array

PRO TIP: THE LET FUNCTION

LET()

Allows you to declare variables, assign values, and use them within formulas

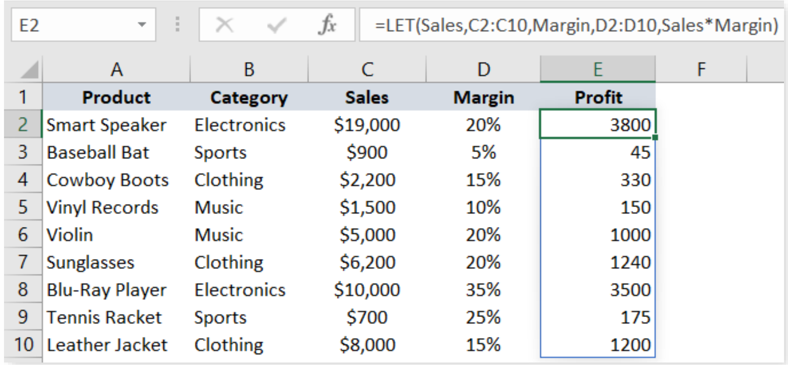
=**LET** (name1, name_value1, calculation_or_name2, *[name_value2]*, *[…]*)

Name of the variable (must begin with a letter)

Value or calculation assigned to the variable

A calculation using the variable, or the name of another variable (optional)

Additional pairs of variable names and values



E2 =LET(Sales,C2:C10,Margin,D2:D10,Sales*Margin)

	A	B	C	D	E	F
1	**Product**	**Category**	**Sales**	**Margin**	**Profit**	
2	Smart Speaker	Electronics	$19,000	20%	3800	
3	Baseball Bat	Sports	$900	5%	45	
4	Cowboy Boots	Clothing	$2,200	15%	330	
5	Vinyl Records	Music	$1,500	10%	150	
6	Violin	Music	$5,000	20%	1000	
7	Sunglasses	Clothing	$6,200	20%	1240	
8	Blu-Ray Player	Electronics	$10,000	35%	3500	
9	Tennis Racket	Sports	$700	25%	175	
10	Leather Jacket	Clothing	$8,000	15%	1200	

*This defines two variables, **Sales** and **Margin**, and multiplies them to return the profit*

PRO TIP: Use **LET** to write clean, efficient, user-friendly formulas

EXTRA BONUS FUNCTIONS

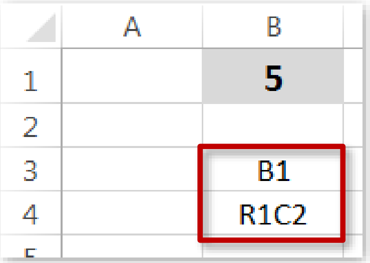
INDIRECT

The INDIRECT function returns the reference specified by a text string, and can be used to change a cell reference within a formula without changing the formula itself

=INDIRECT(ref_text, [a1])

Which cell includes the text that you are evaluating?

Is your text string in **A1** format (1) or **R1C1** format (0)?



	A	B
1		5
2		
3		B1
4		R1C2

ROW(B3) = 3

ROW(INDIRECT(B3)) = 1

ROW(INDIRECT(B4,0)) = 1

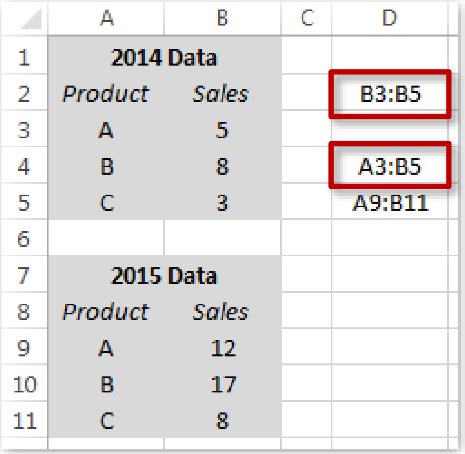
*In the first **ROW** function, Excel returns the row number of cell B3, regardless of what value it contains.*

*When you add **INDIRECT**, Excel sees that cell B3 contains a reference (B1) and returns the row **of the reference***

MAVEN ANALYTICS

INDIRECT (cont)

Let's be real, the INDIRECT function is pretty confusing at first. Here are a few more examples that should give you a sense of how it works and why it can be useful:



	A	B	C	D
1	2014 Data			
2	*Product*	*Sales*		B3:B5
3	A	5		
4	B	8		A3:B5
5	C	3		A9:B11
6				
7	2015 Data			
8	*Product*	*Sales*		
9	A	12		
10	B	17		
11	C	8		

SUM(D2) = 0

SUM(INDIRECT(D2)) = 16

*The sum of "B3:B5" as a **value** doesn't make sense, but the sum of B3:B5 as a **reference** is valid – INDIRECT tells Excel to recognize that the cell you're referring to is a reference, not a value*

VLOOKUP("A", D4, 2, 0) = #N/A

VLOOKUP("A", INDIRECT(D4), 2, 0) = 5

*INDIRECT will tell a VLOOKUP formula to use an array contained **within** a cell, rather than treat the **cell itself** as the array (which returns #N/A)*

HYPERLINK

HYPERLINK creates a shortcut that links users to a document or location within a document (which can exist on a network server, within a workbook, or via a web address)

=HYPERLINK(link_location,[friendly_name])

Where will people go if they click?

How do you want the link to read?

=HYPERLINK(”http://www.example.com/report.xlsx”, “Click Here”)

=HYPERLINK(“[C:\My Documents\Report.xlsx]”, “Open Report”)

=HYPERLINK("#Sheet2!A1”)

PRO TIP:

Use =HYPERLINK("#'"&A2&"'!A1") to jump to cell A1 of the sheet name specified in A2 (note the extra single quotation marks!)