

Excel Magic

This handout contains a variety of functions and tricks that can be used for cleaning and/or analyzing data in Excel. This handout refers to data in an Excel file called "Excel Magic_Data"

Date Functions:

Month-Day-Year (use worksheet called "Dates"):

This is one of my all-time favorite tricks. It works in both Excel and Access. It allows you to grab just one piece of a date. So if you have a series of dates and you want a new field that just gives the year. Or if you want a new field to just list the month.

=Year(Datefield)
=Month(Datefield)
=Day(Datefield)

So if you have 4/3/04, here's what you'll get with each formula:

Year: 2004

Month: 4

Day: 3 (it gives the date, as in the 3rd day of the month)

Weekday:

This works much the same way as the above formula, but instead it returns the actual day of the week (Monday, Tuesday, etc). However the results come out as 1 (for Sunday), 2 (for Monday).

=Weekday(Datefield)

Here's what the answers look like for one week in January:

1/19/2004	2
1/20/2004	3
1/21/2004	4
1/22/2004	5
1/23/2004	6
1/24/2004	7

Note: If you want the 1 value to represent Monday (then 2 for Tuesday, 3 for Wednesday, et), add a 2 on to the formula like this:

=weekday(datefield,2)

Displaying words instead of numbers:

Go to Format > Cells and choose Custom and type "ddd" in the Type box provided. It will display 1 as "Sun", 2 as "Mon", etc. However, the underlying information will remain the numbers. So if you want to base an IF..THEN statement on this field or something like that, your formula would need to refer to the numbers.

DateValue:

If you imported some data and your Date field stayed as text and is not being recognized as a true date (which is necessary for proper sorting), here's how you can fix it. The date has to appear like a real date --- in other words, either 3/4/04 or March 4, 2004 or 4-March-2004 or one of the other recognized date formats. You can tell that Excel is not recognizing it as a date if the text is pushed all the way to the left of the cell. See picture:

Text Version	Date Version
5/5/04	5/5/2004

=DATEVALUE(String)

The String that goes inside the parentheses is the cell where your data starts.

Example: =DATEVALUE(b2)

Date:

If your date is displayed in pieces – month, day, year – or if you want to strip the date portion from a date/time field (see Response Times section below), you can use this DATE function.

This formula assumes you have month, day, year in separate fields:

=DATE(year(a2), month(b2), day(c2))

This formula assumes you have a date/time field:

=DATE(YEAR(A2),MONTH(A2),DAY(A2))

- Note the “DATE” function used here requires you to put the year first, then month, then day. A little counterintuitive .

Datedif:

Useful for calculating ages from birthdates. It gives you the difference between two dates in whatever unit of measure you specify.

=Datedif(Date 1, Date 2, Unit of Measure)

Units of Measure:

“y” --- years

“m” ---months

“ym” ---number of months since the last year

You can use the TODAY() function to refer to today's date. Or you could put a specific date in there (with quotes around it)

Examples:

=Datedif(b2,today(), “y”)

=Datedif(b2, “1/1/2004”, “y”)

Weeknum:

This one requires that you have the Analysis ToolPak installed. It is an add-in for Excel. If the install of Excel was done properly, you should be able to go to the Tools Menu and choose “Add-ins” and then click the check box next to Analysis ToolPak. If that option is grayed out that means you need to re-install Excel.

Weeknum returns the number that corresponds to where the week falls numerically during the year. The formula looks like this:

=Weeknum(celladdress)

Response Times (use worksheet called “time”):

One of the most common things journalists want to do with a date/time field is to calculate response times of local public safety units. To do this, you need to make sure to have full date/time fields for all the key time points you want to compare (i.e. time of 911 call, dispatch time, arrival time, cleared time). Be sure that these have dates for each time, as well, because calls that occur just before midnight might result in an arrival or cleared time occurring on a different date.

Even if you’re not doing response times, a useful formula you might need would be this one to strip the time portion off of a date/time field:

=TIME(HOUR(h4),MINUTE(h4),SECOND(h4))

The best approach for calculating a response time is to convert your time into seconds. Here are the steps you’ll need to do that. (use the worksheet called “TIME” to follow along):

This assumes that you have a date/time field (i.e. “3/31/2013 12:00 PM” or “3/31/2013 14:00”):
=TIME(HOUR(h4),MINUTE(h4),SECOND(h4))*86400

Note: 86400 is the number of seconds in a 24-hour period. So this answer is really representing the time as the number of seconds that have elapsed since midnight.

If you have response times with just a time—no date (i.e. “12:00 pm”), then you can just multiply that by 86400.

To deal with calls that run across midnight (call received in p.m. and the arrival time is in a.m.), we need to be able to handle these differently than the other calls. So we need our formula to be able to check for that.

The simplest would be to have it look to see if the receive date is different than the arrive date. However, our fields have both date AND time. So it might help if we add new fields that just hold the dates.

So we’ll create “RECEIVE DATE” AND “ARRIVE DATE” fields and populate them using these formulas:

=DATE(YEAR(H4),MONTH(H4),DAY(H4))

=DATE(YEAR(J4),MONTH(J4),DAY(J4))

- Note the “DATE” function used here requires you to put the year first, then month, then day. A little counterintuitive .

Of course, if you’re feeling confident, you could build that date function into the formula below. It would just make a really long and complex formula.

Now we can calculate the response time – the difference between the receive time and the arrive time, and display our answer in minutes.

Here’s the formula, then I’ll explain:

`=(IF(N4=O4, M4-K4, (86400-K4)+M4))/60`

This criteria portion of this IF statement is “N4=O4” – it’s looking to see if the “receive date” and “arrive date” are on the same day (if not, that’s an indicator that this runs across midnight).

If that’s true, it subtracts M4-K4 (arrive time seconds minus receive time seconds)

If the criteria is false, it subtracts 86400 (number of seconds in a day) from K4 (the receive time) and then adds the arrive time. This strange formula puts the receive time and the arrive time into the same time frame to make it possible to subtract without getting a negative number.

Finally, we have this whole formula surrounded by parentheses and then divide by 60 off the end. This converts the answer from seconds to minutes.

Displaying data as a calendar: (use worksheet called “Calendar”)

You can use Weeknum and Weekday (listed above) in conjunction with Pivot Tables to display data in a sort of calendar form. This would be useful if you’re looking for patterns in your data based on the calendar.

To do that, you need to add fields to your data with WeekNum and WeekDay corresponding to the date in that field. Then create a Pivot Table, with WeekNum in the Row, WeekDay in the Column and whatever field you want to count or sum in the Data box. (I found that you need to leave the WeekDay output as 1, 2, 3, etc., so that it will display in the proper order. I tried to have them display as “Mon”, “Tues”, etc and it wouldn’t put them in order)

Text or String Functions:

(use worksheets called “split names” or “split address”)

These are extremely handy tools that you can use for data cleanup (particularly splitting names) or during analysis. They allow you to grab only a piece of the information in a field based on certain criteria. These functions are also available in Access, however there are a couple slight variations in syntax. Once you know one, learning the other is a breeze.

LEFT: This tells the computer to start at the first byte on the left side of the field. Then we have to tell it how many bytes (or characters) to take.

Syntax: LEFT(celladdress, number of bytes to take)

Example: LEFT(B5, 5) --- this will extract the first 5 characters of the contents of cell B5

MID: To use this function, you have to tell the computer which cell to do its work, where to start and where to stop. If you want to take everything that remains in the field, just put a really big number in that will likely encompass all possibilities.

Syntax: MID(celladdress, byte number to start at, number of bytes to take)

Example: MID(B5,10,4) --- this will start at the 10th byte and take 4 bytes.

SEARCH: This works as a sort of search tool to tell the computer to either start or stop taking a “string” at a certain character (or space). This is how we can tell the program to split a name field at the comma, for example. For this type of work, it is used in conjunction with the MID function. The character you want to find should be enclosed in quotes.

Syntax: SEARCH(“character we want to find”, celladdress)

Example: SEARCH(“,”,B5)

You can combine this with Mid to explain that you either want to start or stop at a certain character (even if the character isn’t located at the same byte in every record).

EXAMPLE: MID(b5, search(“,”, b5), 100)

**the above example uses the search function to find the “start” position, then tells the computer to take 100 bytes from there.

EXAMPLE: MID(b5, 10, search(“,”, b5))

**the above example uses search to find the “end” position.

**Note: If you don’t want to include the character that you searched for in your result, use a –1 or +1 just after the search phrase to either go back a space (-1) or move forward and start a space farther (+1). Here’s an example that will start at the comma, then move one space forward and take 100 bytes from there:

=mid(b5, search(“,”,b5)+1, 100)

There is also a **RIGHT** function, which starts at the first byte on the right side of the field and then you can tell it how many bytes to take. (it isn’t as useful as the others, however)

Trick for splitting apart city and state when it’s not delimited

(use worksheet called “citystate”)

This trick is only going to work in specific circumstances, but it’s one you might encounter with some frequency. Here’s the deal...you’ve got a spreadsheet that has a column containing both the city name (or perhaps a county name) and a two-digit state abbreviation but there isn’t a comma separating the two items so it’s not easy to parse.

You can use the LEN function to determine how long the full string is and then subtract 2 digits to find out what byte position that last space is at. (since that’s the byte position you want to use for splitting the info).

So in a new column, use this formula and copy it down:

=len(a2)-2

Check your numbers on a few examples to make sure it’s hitting the right position. Then you can use that number you just created — assume that the new set of numbers are stored in the B column.

To grab the city name:
=LEFT(a2,b2)

See how I substituted “b2” instead of putting the search(“”, a2) like we did in the example above?

Then you can grab the state abbreviation either by using:
=RIGHT(A2,2)
OR
=MID(A2,B2,2)

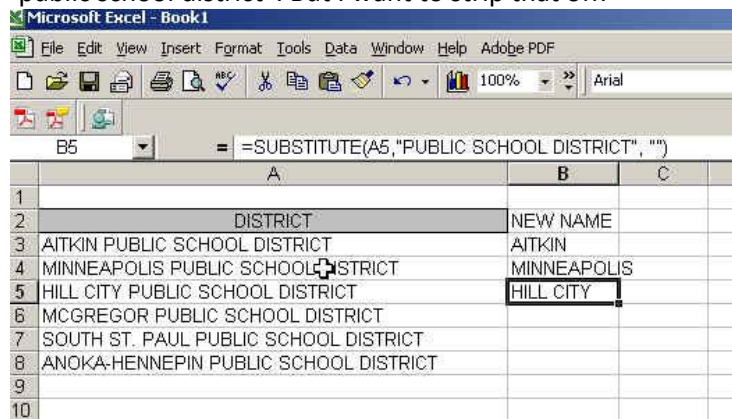
Other text functions:

SUBSTITUTE(cell, oldtext, newtext): Allows you to mass replace (or elimination) of a specific word or phrase in a column.

One way this can be super useful is if you have data with either a leading or trailing non-space whitespace. In other words, you’ve tried to TRIM() the field and you still have a space.

You can use Substitute to have it look for that non-space whitespace --- char(160) – and replace it with nothing.
=substitute(a1,char(160),””)

Another example.... I have a list of school districts and the names of the schools all end with “public school district”. But I want to strip that off.



The screenshot shows a Microsoft Excel spreadsheet with the following data:

	A	B	C
1			
2	DISTRICT	NEW NAME	
3	AITKIN PUBLIC SCHOOL DISTRICT	AITKIN	
4	MINNEAPOLIS PUBLIC SCHOOL DISTRICT	MINNEAPOLIS	
5	HILL CITY PUBLIC SCHOOL DISTRICT	HILL CITY	
6	MCGREGOR PUBLIC SCHOOL DISTRICT		
7	SOUTH ST. PAUL PUBLIC SCHOOL DISTRICT		
8	ANOKA-HENNEPIN PUBLIC SCHOOL DISTRICT		
9			
10			

The formula bar shows the formula: =SUBSTITUTE(A5,"PUBLIC SCHOOL DISTRICT", "")

Here’s the formula I used in the above example:
=SUBSTITUTE(a3, “PUBLIC SCHOOL DISTRICT”, ””)

In the above example I’m leaving the “newtext” part of the formula blank because I don’t want to replace the phrase with something else. If you wanted to change it — perhaps you want it to say, “Schools” — then you could put that within that last set of quotes.

The function is very specific. For example it won’t replace the phrase “PUBLIC SCHOOL DIST” because it’s not an exact match.

EXACT(text1, text2): (use worksheet called “Exact”) Compares two strings to see whether they are identical. This is great for if you are trying to line up two sets of lists. Let’s say each contains the 50 states, so you want to align them by the name of the state (which appears in both lists). It returns FALSE if the two items are not identical.

=EXACT(E1, F1)

E	F	G
minnesota	minnesota	TRUE
new jersey	virginia	FALSE
virginia	new jersey	FALSE
delaware	delaware	TRUE

REPT(text, number): (use worksheet called “Rept”) This one is kind of interesting. It repeats the given text whatever number of times you tell it. The most interesting use of this I found is to generate a sort of bar chart on the fly. So for example, let’s say you have a list with totals of something in column B.

You could have it create bar charts using the pipe “|” character based on the total number, like this:

=REPT(“|”, b2)

When you copy this down to the remaining rows you’ll see it create a bar for each line.

	A	B	C	D
1				
2		Per Month		
3	Company A	10		
4	Company B	4		
5	Company C	6		
6				
7				

LEN(text): Returns the length in number of bytes.

PROPER(text): Converts the data in the cell to proper case. LOWER and UPPER are also available.

Logical Functions:

IF Statements:

(use worksheets called “BasicIF”, “More BasicIF”)

These are one of several LOGICAL functions that are available in Excel. It’s an extremely powerful tool for a variety of tasks, most notably for assigning categories to your data based on certain criteria and for some data cleanup functions that require looking for patterns. Essentially they allow you to do one thing if your criteria is true, and another thing if your criteria is false. Later, we’ll talk about nested IF functions that allow you to use multiple criteria.

A basic IF statement consists of:

- 1) What we’re going to measure as being either true or false
- 2) What to do if it’s true
- 3) What to do if it’s false

=IF(criteria, true, false)

So here's an example from a list of football games. We want to identify whether the visiting team or the home team won the game.

	A	B	C	D	E	F	G	H
1	Date	WeekNum	VisitAbbrev	HomeAbbrev	VisitScore	HomeScore	Winner	
2	9/4/03	1	NYJ	WAS	13	16	=if(e2>f2,"visit","home")	
3	9/7/03	1	ARI	DET	24	42		
4	9/7/03	1	DEN	CIN	30	10		
5	9/7/03	1	IND	CLE	9	6		

This formula will insert the word "Visit" in the G cell if the measurement is true and will insert "Home" if the measurement is false.

You can also have it grab information from other cells, instead. This will put the VisitAbbrev (i.e. "NYJ") in the H column if the measurement is true and will grab the HomeAbbrev if it's false.

	A	B	C	D	E	F	G	H	I
1	Date	WeekNum	VisitAbbrev	HomeAbbrev	VisitScore	HomeScore	Winner	Win Team	
2	9/4/03	1	NYJ	WAS	13	16	home	=if(e2>f2,c2,d2)	
3	9/7/03	1	ARI	DET	24	42	home		
4	9/7/03	1	DEN	CIN	30	10	visit		
5	9/7/03	1	IND	CLE	9	6	visit		

Using IF to copy down blank columns:

(use worksheet called "Copy down")

I use this quite frequently when I get data that lists a team name as a title, then all the players or all the game dates below that. But I want to apply the team name to each record. Some of the Census products that are already summarized and formatted have this problem as well.

The trick is that you need to have a pattern to follow. In the example below, the pattern is that the B column is always blank on the lines where the team name is listed. And it's not blank anywhere else.

	A	B	C	D
1	Name	Position		
2	Arizona Cardinals			
3	Starks, Duane	DB	=if(b2="",a2,c2)	
4	Stone, Michael	DB		
5	Ransom, Derrick	DT		

So this formula is going to look to see if the B cell is blank:

=IF(b2="", a2,c2)

Then it's going to put the contents of A2 (in this case, "Arizona Cardinals") in the field if it finds it to be true. If it's not true, it looks to the cell directly above (c2) to essentially copy down the team name.

Combining other functions:

Now that you know the basics of an IF statement, you can jazz it up with all kinds of other functions. You just place the function as either the criteria, the true part or the false part. Of course, you can use multiple functions in the same IF statement if necessary.

Examples (I've just made these up!):

If a date (located in b2) is for a Monday, then put the word Monday in a new cell, otherwise do nothing:

=if(weekday(b2)=2, "Monday", "")

If a date (located in b2) is equal to another date (located in c2), then put the word "Same" in the new field, otherwise calculate the difference in months:

=if(b2>c2, "Same", datedif(b2,c2, "m"))

Using a wildcard search:

You can use the SEARCH function to look for a word or symbol contained within other text, however it gets a little tricky to make it work properly. You have to add the ISERROR function. If you want to get in this deep, I recommend checking out the help file on these functions.

Otherwise, here's a quick hit to get you started:

	A	B	C
1			
2			
3			
4		Arlington, Texas	X
5		Chicago, Illinois	
6		Dallas, Texas	X
7		Fort Worth, Texas	X
8		Reston, Virginia	
9			

This example assumes you have a list of cities and states and you want to flag all of the ones that are in Texas. In this case, the state name is written out in full.

So if it finds Texas, this formula instructs it to put an X in the C column, otherwise leave it blank.

=IF(ISERROR(SEARCH("*Texas*",B4,1)>0)=FALSE, "X", "")

The criteria part of this stretches from the ISERROR all the way to the FALSE. The ISERROR is necessary because it will give you an error message if it doesn't find the word. It's the only way you can instruct the computer to do something in the false portion of your answer (even if that means just leaving it blank).

The following portion:

SEARCH("*Texas*", b4, 1)

If used alone, this portion will return a 1 if it finds the search term and an error message if it doesn't. So then you need to add the IF portion to give it two options. By adding the ISERROR and the =FALSE, you can sidestep the error message.

Nested IF statements:

(use worksheet called "Nested IF")

Once you understand basic IF statements, then you can really do some complex stuff with nested IF statements. In other words, we can say: if something is true, then check to see if something else is true. If both are true, do this. If the first is true, but the second is not, then do this. If both are false, then do this.

The sheet in ExcelTricks called "Nested IF" has data from Minnesota's gay marriage battle. It has one record for each state house district. Then in columns B through G we have results from a 2012 statewide ballot measure calling for a ban on gay marriage. The last three columns indicate the legislator in that district in 2013, their party affiliation (DFL=Democrat) and how they voted on a bill in 2013 that allowed gay marriage. (Ultimately the bill was signed into law).

The goal here was to find legislators who were not in sync with their district on this issue. Political analysts were saying these would be the legislators who would be targeted by the opposing party in the next round of elections.

So there are two things we can do here:

- 1) First, let's identify which districts approved the ballot measure (PctYes>.5) or not.
- 2) Second, let's identify which legislators voted the opposite of their constituents.
- 3) Then, of the ones who were opposite of their constituents, which districts either passed or opposed the 2012 ballot measure by a large margin (60% or more)

legdist	NoVotes	YesVotes	EstBlanks	PctYes	PctNo	Legislator	LegisParty	LegisVote	Ballot Result	Opposites?	How wide?
17B	7442	12218	238	0.614	0.374	Rep. Mary Sawatzky	DFL	No	Y		
24B	7498	10561	187	0.579	0.411	Rep. Patti Fritz	DFL	No	Y		
02A	7957	12226	269	0.598	0.389	Rep. Roger Erickson	DFL	Yes	Y		
03A	11139	11830	305	0.508	0.479	Rep. David Dill	DFL	Yes	Y		
03B	12140	11113	256	0.473	0.516	Rep. Mary Murphy	DFL	Yes	N		
04A	10224	9007	333	0.460	0.523	Rep. Ben Lien	DFL	Yes	N		
04B	7765	11926	318	0.596	0.388	Rep. Paul Marquart	DFL	Yes	Y		
05A	9700	10620	255	0.516	0.471	Rep. John Persall	DFL	Yes	Y		
05B	9008	12467	219	0.575	0.415	Rep. Tom Anzelc	DFL	Yes	Y		
06A	9865	11181	243	0.525	0.483	Rep. Carly Melin	DFL	Yes	Y		
06B	10747	12254	269	0.527	0.462	Rep. Jason Metsa	DFL	Yes	Y		
07A	17011	7140	266	0.292	0.697	Rep. Thomas Huntley	DFL	Yes	N		
07B	12809	8454	221	0.394	0.596	Rep. Erik Simonson	DFL	Yes	N		
10A	8886	12336	197	0.576	0.415	Rep. John Ward	DFL	Yes	Y		
10B	8312	14317	238	0.626	0.363	Rep. Joe Radinovich	DFL	Yes	Y		
11A	9748	10751	175	0.520	0.472	Rep. Mike Sundin	DFL	Yes	Y		
11B	6841	12016	167	0.632	0.360	Rep. Tim Faust	DFL	Yes	Y		
12A	7748	14117	255	0.638	0.350	Rep. Jay McNamar	DFL	Yes	Y		
14B	10920	8161	227	0.423	0.566	Rep. Zachary Dorholt	DFL	Yes	N		
17A	6724	13312	392	0.652	0.329	Rep. Andrew Falk	DFL	Yes	Y		
19A	11508	10459	222	0.471	0.519	Rep. Clark Johnson	DFL	Yes	N		
19B	12388	8562	247	0.404	0.584	Rep. Kathy Brynaert	DFL	Yes	N		
20B	12522	9448	268	0.425	0.563	Rep. David Bly	DFL	Yes	N		
25B	11288	10305	308	0.469	0.514	Rep. Kim Horton	DFL	Yes	N		
26A	10411	8333	345	0.437	0.545	Rep. Tina Liebling	DFL	Yes	N		
27A	8191	12723	248	0.601	0.387	Rep. Shannon Savick	DFL	Yes	Y		
27B	7878	10853	260	0.571	0.415	Rep. Jeanne Poppe	DFL	Yes	Y		
28A	11590	9457	281	0.443	0.543	Rep. Gene Polowski, Jr.	DFL	Yes	N		
36B	11508	10801	342	0.477	0.508	Rep. Melissa Hortman	DFL	Yes	N		

Step 1 – In the “Ballot Result” column, let’s identify whether ballot measure passed (Y) or not (Y) or if there was a tie (TIE)

We’ll need a nested IF statement to do that. Remember that an IF statement is made up of 3 parts – the criteria (Excel calls it the “logical test”), what to do if it’s true, and what to do if it’s false. When you “nest” an IF statement you just drop it into either the true spot or the false spot of the first IF statement.

So in this example, I want to see if the PctYes was greater than 50%, and put “Y” in our column if that’s true. If that’s false, then I want to check for a tie and put “tie” in if it’s true, and put “N” in if that’s false.

=IF(CRITERIA1, TRUE, IF(CRITERIA2, TRUE, FALSE))

Exact formula for this one:

=IF(e7>.5, “Y”, if(e7=.5, “tie”, “N”))

Step 2 – let’s now identify which district have “opposites” – the legislator voted one way on the gay marriage bill in 2013 and his/her constituents went the other way on the 2012 amendment. We’ll use the field we just created (ballot result) and “LegisVote,” which shows how the legislator voted in 2013 as a yes or no.

The tricky thing about this one is that “Y” on the amendment and “No” on the Legislator’s vote actually mean the same thing – both are opposed to gay marriage.

I want a new field that says either “opposite”, “both opposed” or “both in favor”

We’re going to use 3 IF statements for this one. There are different ways to set this up, but it works generally the same way.

```
=if(I7= “no”, if(J7=“Y”, “both opposed”, “opposite”), if(j7=“N”, “both in favor”, “opposite”))
```

Here’s how to interpret this. First it looks to see if there’s a “No” in the LegisVote column (I). If true, then it looks to see if there’s a “Y” in the Ballot Result column (J). If that’s true that means both are opposed. (In other words, we had true on first IF and true on second IF). If the Ballot result column is NOT “Y”, then it’s going to insert “opposite.” (in other words, true on first IF, but false on second IF).

The third IF statement is actually the FALSE portion of the first IF statement. So that one won’t even kick in unless I7=“NO” (our first criteria) is false – in other words, the legislator voted “yes” So in that scenario, the legislator voted “yes” (it failed the first IF statement), so it skips past the second IF statement and goes to the 3rd IF statement to see if the ballot result (J) is “N”. If that’s true, then it says “both in favor”. If it’s false, then it says “opposite”(legislator voted “yes” and constituents approved the ban).

An alternative way of doing this would be to use the **AND() function**. This allows you to have 2 criteria in the same IF statement. In this, we’re going to nest 2 IF statements, both using the AND function.

Here’s how the AND fits into an IF statement:

```
=IF(AND(criteria1, criteria2), true, false)
```

Here’s the formula we’ll use for this one:

```
=if(AND(I7=“yes”, j7=“N”), “both in favor”, if(AND(I7=“no”, J7=“Y”), “both opposed”, “opposite”))
```

This first looks to see if the legislators and constituents are both in favor, if that’s false then it looks to see if they are both opposed. And then if that’s ALSO false (now we’ve got 2 false ifs), then it says there’s an “opposite” going on.

SUMIF and COUNTIF Functions:

(use worksheets called “SumIF” and “CountIF”)

If you’ve got a long list that you want to essentially do subtotals for, this is a way you can do it without moving it to Access or doing a lot of repetitive typing of formulas or using Pivot Tables. This would be a better option than Pivot tables if you only want to do subtotals on a sub-set of your data.

The example uses a list of player salaries for the NBA. I want to know the total for each team.

	A	B	C	D	E
1	Year	Player	Team Name	Position	Salary
2	2003-04	Abdul-Wahad, Tariq	Dallas Mavericks	F-G	\$6,187,500.00
3	2003-04	Abdur-Rahim, Shreef	Atlanta Hawks	F	\$13,500,000.00
4	2003-04	Alexander, Courtney	New Orleans Hornets	G	\$2,179,000.00
5	2003-04	Allen, Malik	Miami Heat	F	\$638,679.00
6	2003-04	Allen, Ray	Seattle SuperSonics	G	\$13,500,000.00

The formula requires three pieces:

=sumif(range to evaluate, criteria, range to sum)

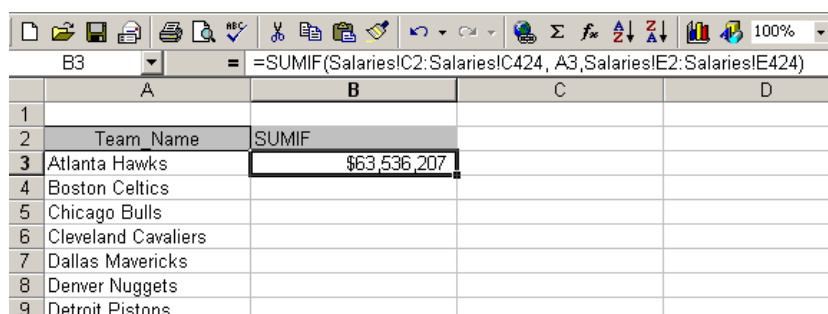
In this example, let's say we want to subtotal the Dallas Mavericks salaries. So the range to evaluate will be the C column (c2:c424). The criteria would be "Dallas Mavericks" (put it in quotes because it's a text string). And the range to sum would be the E column (e2:e424).

Here's the full formula:

=sumif(c2:c424, "Dallas Mavericks", e2:e424)

To do this for all teams in one sweep, I put a list of the teams in a separate worksheet and then "link" the formulas between the two worksheets. The worksheet with the player-by-player salaries is named "Salaries" and the team names are in the A column of my new worksheet. Since the players are in alpha order and not according to team, it's necessary to "anchor" the formula that adds the salaries together. Here's what the formula ends up looking like:

=sumif(Salaries!C2:Salaries!C424, A3, Salaries!\$E\$2:Salaries!\$E\$424)



	A	B	C	D
1				
2	Team Name	SUMIF		
3	Atlanta Hawks	\$63,536,207		
4	Boston Celtics			
5	Chicago Bulls			
6	Cleveland Cavaliers			
7	Dallas Mavericks			
8	Denver Nuggets			
9	Detroit Pistons			

There's a similar function called COUNTIF that will do the same thing, only it will count the number of instances rather than adding numbers together. The formula is a bit shorter:

=countif(range to evaluate, criteria)

I see a couple possible ways to use this. The first is to simply return a single number that counts how many records meet a certain criteria. For example, in the County Business Patterns data, I want to know how many counties have at least one business that employs 1,000 or more people. There are separate columns listing the numbers of businesses based on employee number ranges: 1 to 4, 5 to 9, etc. The last column is for 1000 or more employees. I could use this formula to count how many counties have at least one business in this range:

=COUNTIF(\$N\$3:\$N\$90, ">0")

You need to use the quote marks if you want to do greater than, less than or something like that. If you just want to find how many records have a specific number in the N column, then you don't need the quotes.

Other useful stuff....

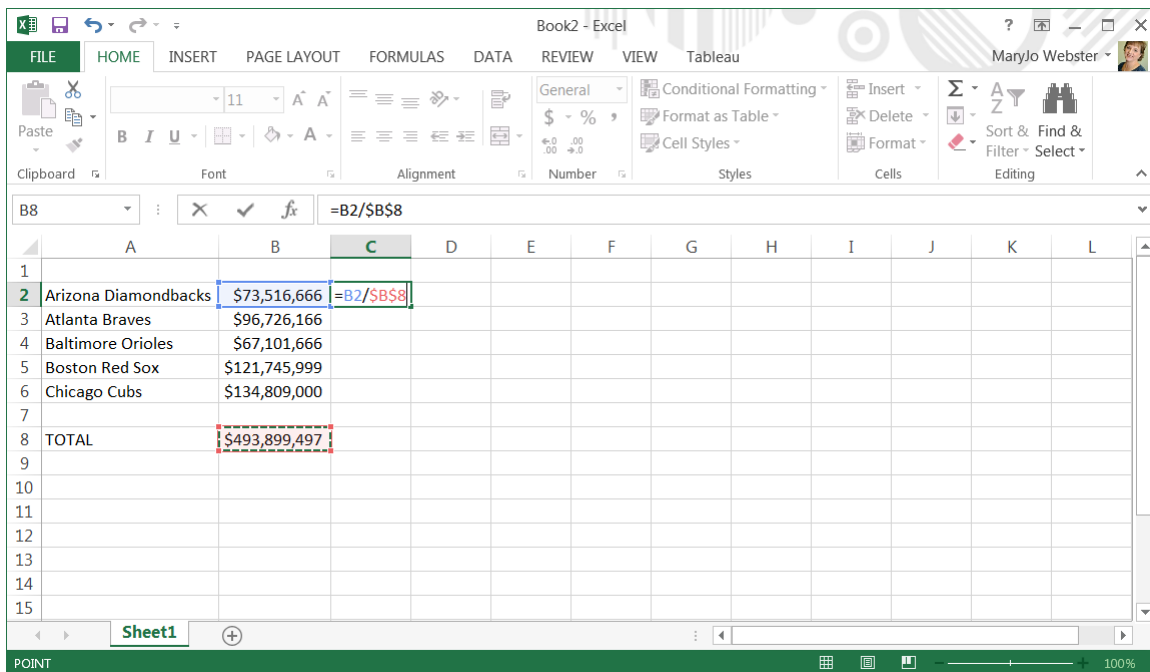
Anchors:

When you need to use an anchor (\$) in a formula, here's a quick way to insert it without a lot of typing. So here's an example. Let's say you need to do a percent of total, like in the example below. Type the formula without the anchors

=b2/b8

And then push the F4 key. It will insert the \$ to lock the B8 cell. This locks it so that it won't change if you copy the formula down, or copy across.

A bit more about anchors.....If you want to allow the column to change but not the row, you would only use the anchor in front of the number. If you want to allow the row to change, but not the column, then you only use the anchor in front of the column letter.



Rank:

This is a more sophisticated way to rank your records and to account for ties.

=RANK(This Number, \$Start Range:\$End Range\$, Order)

- This Number should be the cell where your data starts.
- Start Range should be the cell where your data starts. Anchor with dollar signs.
- End Range should be the last cell of your data. Anchor with dollar signs.
- Order is either a 1 (smallest value will get assigned #1) or a 0 (largest value will get assigned #1).

Example: =RANK(B2,\$B\$2:\$B\$100,1)

PercentRank:

Returns the rank — or relative standing - within the dataset as a percentage. So for example, if you had a list of the payrolls for all of the Major League Baseball Teams, you could do a percent rank on the payroll to find out which team (the Yankees, of course) have the greatest percentage of the total.

=PERCENTRANK(array, x, significance)

Array: The range of data that you want to compare each item to

X: the value for which you want to know the percent rank

Significance: an optional value that allows you to set the number of digits

Example:

=PERCENTRANK(\$a\$2:\$a\$30, a2, 2)

Also check out **PERCENTILE** and **QUARTILE** functions in the Help file.

Round:

=ROUND(cell, num_digits): For this one you tell it which cell to do it's work on and then the number decimals you want to round to. For the num_digits you can use something like this.

These examples show how it would round the number 1234.5678

- 0 puts it to the nearest integer (1235)
- 1 goes to one decimal place (1234.6)
- -1 goes to the nearest tenth (1230)
- -2 to the nearest hundredth (1200)
- -3 to the nearest thousandth. (1000)

Lookup Tables:

(use worksheets called "lookups", "lookup2" and "lookup3")

The VLOOKUP and HLOOKUP functions allow you to use Excel more like a relational database program. So if you haven't made the leap to Access yet, here's how you can get more functionality out of Excel.

Both functions are useful for cases where you have data that relates to another chunk of data, with one field in common. They work best if you have the data in the same workbook, but it can be on separate worksheets. This might be the list of 50 states with current Census estimate data in one worksheet and the same list with last year's data in another worksheet. Or it might be a one-to-many relationship where you have a list of cell phone calls and another table that groups the time of day into categories (such as morning, evening and afternoon).

The difference between VLOOKUP and HLOOKUP is that VLOOKUP will troll through your lookup table vertically (all in one column). HLOOKUP goes through it horizontally, or all in one row.

To demo this, we'll use a very simple example. One worksheet has data from the Census County Business Patterns, but each record is only identified by the county FIPS number. I want to add a field that shows the county name. A second worksheet has the names associated with the FIPS numbers.

	A	B
1		
2	Code	County
3	001	Aitkin
4	003	Anoka
5	005	Becker
6	007	Beltrami
7	009	Benton
8	011	Big Stone
9	013	Blue Earth
10	015	Brown
11	017	Carlton
12	019	Carver
13	021	Cass
14	023	Chippewa
15	025	Chisago
16	027	Clay
17	029	Clearwater
18	031	Cook

VLOOKUP requires that the field you're matching on is the farthest left column of the lookup table, like in my example pictured here. (Below I'll show you how to use different functions if your table is not set up this way).

In this example, my Business Patterns data is in one worksheet and this lookup table is in a worksheet called "Lookup2". Our formula will need to reference that name, so it's a good idea to name your worksheets when you do this.

Here's the structure for VLOOKUP:

VLOOKUP(cell, range of lookup table, column number, range_lookup)

The cell is the first cell in your data table. In this case it would be the cell containing the first FIPS number I want to look up.

Range of lookup table is the upper left corner of your lookup table to the lower right corner, encompassing all fields. In the example pictured above, it would be worded like this:

Lookup2!\$A\$3:\$B\$89

"Lookup2!" is how we refer to the other worksheet, then you need to anchor (\$) the starting cell (A3) and ending cell (B89).

The column number refers to the column number of your lookup table that you want to return in your data. In this case, I want column 2, which contains the name of the county.

For range_lookup you either put TRUE or FALSE. True will first search for an exact match, but then look for the largest value that is less than your data value. FALSE will only look for an exact match. In this case we want to use FALSE. You'll see below when you would want to use True.

So here is our final formula:

=VLOOKUP(B3, Lookup2!\$A\$3:\$B\$89,2, FALSE)

Name your lookup: You can simplify this formula by naming your lookup table. Highlight the cells in your lookup table, in this case A3 to B89. Go to the Insert Menu and choose Name, then choose Define. Then type in a name (all one word). For this example, let's say I called it "FIPSlookup".

Then you can change your formula to this:

=VLOOKUP(B3, FIPSlookup, 2, FALSE)

MATCH and INDEX:

As I mentioned above, there is another option if your lookup table is set up differently. Let's say the FIPS table starts with the FIPS number and state name in the first two columns, then has the county FIPS number in the 3rd column. (this is the worksheet "lookup3") Obviously VLOOKUP won't work because of the placement of that county FIPS column.

Instead you can use a combination of INDEX and MATCH functions. Let's break it apart first to see how it works. Index will go to the data range specified and return the value at the intersection of the row number and the column number that you provided to it. So, using this alone requires that we provide specific column and row numbers.

	A	B	C	D
1				
2	State	FIPS	Code	County
3	MN	27	001	Aitkin
4	MN	27	003	Anoka
5	MN	27	005	Becker
6	MN	27	007	Beltrami
7	MN	27	009	Benton
8	MN	27	011	Big Stone
				Blue
9	MN	27	013	Earth
10	MN	27	015	Brown
11	MN	27	017	Carlton
12	MN	27	019	Carver
13	MN	27	021	Cass
14	MN	27	023	Chippewa
15	MN	27	025	Chisago

In this example, pictured here, we could use this formula to get it to return "Becker", which is in the 3rd row of the table and the fourth column.

=INDEX(Lookup3!\$A\$3:\$D\$89, 3,4)

But when trying to match this back to the big data table, we need more flexibility. So instead of hard-coding the row number, we're going to drop the MATCH function into its place.

=MATCH(Lookup3!\$C\$3:\$C\$89, FALSE)

So this is going to the C column and by setting FALSE, we are saying we want an exact match.

Here's the final formula:

=INDEX(Lookup3!\$A\$3:\$D\$89, MATCH(B3, Lookup3!\$C\$3:\$C\$89, FALSE),4)

VLOOKUP for inexact match:

I'm going to steal an example from a Sarah Cohen handout from several years back. This one uses cell phone records for public officials that include the time of the call. She wants to flag each of the records into categories based on whether they were likely working at the time the call was made. Here's the lookup table that she created:

Time	AtWork?
12:00 AM	Doubtful
6:00 AM	Tossup
8:00 AM	Likely
5:30 PM	Tossup
7:30 PM	Doubtful

You can use VLOOKUP to check each time in the cell phone log against this table. If it doesn't find an exact match, it will give you back the value corresponding to the next lowest value. So, if the time is 6:30 pm, it will match to 5:30 pm Tossup.

The formula would be essentially the same as above, but you use TRUE as the range_lookup.

So this example might look like this:

=VLOOKUP(b4, NewSheet!\$A\$2:\$B\$6, 2, True)

Copying down a single date:

Excel's wonderful feature of copying down (or across) formulas becomes a bit of a nightmare when you simply want to copy down the same date. Excel will think you want to go on to the next day, then the next, and the next, etc. Here's the trick for disabling that:

****Hold down the Control (Ctrl) key while dragging/copying down the first instance of the date.**

Using column names instead of cell addresses:

Are you sick of typing cell addresses? You can set up your worksheet so that the headers you've typed for each column can be used as cell addresses in your formulas instead. Here's how it works.... First thing to do is make sure your headers are all filled out, that they are single words (no spaces, no punctuation), and that they are stored on the first line of your worksheet. Next, highlight all of your data (my favorite way to do this is to put your cursor somewhere in your dataset and hit Control-Shift-Asterisk).

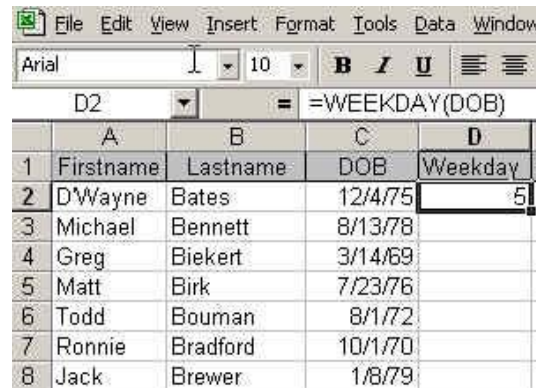


Directions for Office 2003 and earlier: Then go to the Insert menu and choose Name, then choose Create. It will bring up a dialog box called "Create Names" where you should make sure that **ONLY** the "Top row" choice is checked.

Directions for Office 2007: Go to the Formulas ribbon and look for "Name Manager" and a button that says "Create names from selection." In the dialog box that comes up make sure that **ONLY** the "top row" choice is selected.

Directions for both versions:

Once this is set you can use your field names instead of cell addresses. So for example, in our list of football players and their dates of birth, we could calculate the WEEKDAY function (see image below) using "DOB" instead of C2 in our formula. Of course, this would be much more useful if we have really complicated formulas (like the IF...THEN formulas) with lots of cell addresses that tend to get confusing.



	A	B	C	D
1	Firstname	Lastname	DOB	Weekday
2	D'Wayne	Bates	12/4/75	5
3	Michael	Bennett	8/13/78	
4	Greg	Biekert	3/14/69	
5	Matt	Birk	7/23/76	
6	Todd	Bourman	8/1/72	
7	Ronnie	Bradford	10/1/70	
8	Jack	Brewer	1/8/79	

Understanding Errors:

#DIV/0! : This almost always means the formula is trying to divide by zero or a cell that is blank. So to fix this, first check to make sure that your underlying data is correct. In many cases, you will have zeros. For example, the number of minority students in some schools in Minnesota might be zero, so I have to use an IF statement whenever trying to calculate the percentage of minority students. Here's how I get around the error, assuming the number of minority students is in cell B2 and the total enrollment is in C2. If the number of minority students is greater than zero, it does the math. Otherwise it puts zero in my field.

=if(b2>0, b2/c2, 0)

#N/A: This is short for "not available" and it usually means the formula couldn't return a legitimate result. Usually see this when you use an inappropriate argument or omit a required argument. Hlookup and Vlookup return this if the lookup value is smaller than the first value in the lookup range.

#NAME?: You see this when Excel doesn't recognize a name you used in a formula or when it interprets text within the formula as an undefined name. In other words, you've probably got a typo in your formula.

#NUM!: This means there's a problem with a number in your formula (usually when you're using a math formula).

#REF!: Your formula contains an invalid cell reference. For example, it might be referring to a blank cell or to a cell that has since been deleted.

#VALUE!: Means you've used an inappropriate argument in a function. This is most often caused by using the wrong data type.

Tableau Reshaper Tool (only works in Excel 2007 and newer):

Download here and follow the directions to install in Excel :

<http://kb.tableausoftware.com/articles/knowledgebase/addin-reshaping-data-excel>

This is a great tool for “normalizing” data, whether you plan to put it into Tableau Public (visualization software) or not.

Let’s start with the worksheet called “Reshaper1.” This has enrollment data from the University of Minnesota, broken down by race, gender, ethnicity and residency status. For a visualization, like Tableau, or even some analysis purposes, it would be better to have the data lined up with one item (or group total) per line – and keep the grand total attached to each group. Like this:

Year	GrandTotal	Group	Value
1997	32342	Men	15470
1997	32342	Women	15872
Etc.			

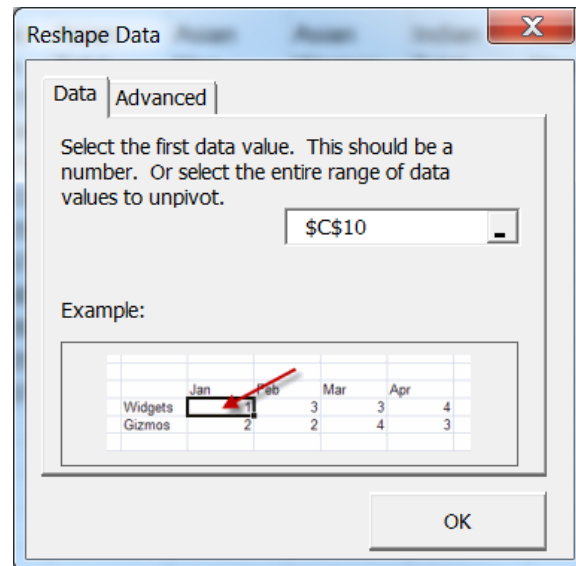
Tableau Reshaper is perfect for this.

First, a little prep work to make this work the best.

- Make sure the headers that are on your columns (in this case, the group names) are presented EXACTLY as you want them to appear in the final data.
- Make sure that the columns you want to convert are all on the right side of your spreadsheet and the columns that you want attached to each row are all on the left side.

Then to make the reshaper tool work, you put your cursor on the first cell that you want converted (notice on the screen capture above, my cursor is on C10 – the first piece of data I want to put in rows).

Finally, go to the Tableau menu (which was added to your menu options when you installed it) and choose “Reshape Data”. It will ask what cell you want to start with – and since you put your cursor there previously, it should guess correctly.



For this one, use the “Reshaper2” worksheet:

Below is an image of data from one of the health insurance exchanges set up under the Affordable Care Act. Each row is an insurance product offered in a particular rating area (geographic area). The premium costs are listed by age, going across the columns (age 0-20, 21, 22, 23, etc)

	Company	Metal	RatingArea	RateArea	0-20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
1	All Savers	Silver	3	Colorado3	189.13	297.84	297.84	297.84	297.84	299.03	304.99	312.14	323.75	333.28	338.05	345.20	352.35	356.81	361.58	363.96	366.34
2	All Savers	Silver	7	Colorado7	186.45	293.62	293.62	293.62	293.62	294.80	300.67	307.72	319.17	328.56	333.26	340.31	347.36	351.76	356.46	358.81	361.16
3	All Savers	Silver	8	Colorado8	196.75	309.84	309.84	309.84	309.84	311.08	317.28	324.72	336.80	346.71	351.67	359.11	366.54	371.19	376.15	378.63	381.11
4	All Savers	Silver	9	Colorado9	245.78	387.06	387.06	387.06	387.06	388.61	396.35	405.64	420.73	433.12	439.31	448.60	457.89	463.70	469.89	472.98	476.08
5	All Savers	Silver	10	Colorado10	242.69	382.19	382.19	382.19	382.19	383.72	391.36	400.54	415.44	427.67	433.79	442.96	452.13	457.86	463.98	467.04	470.09
6	Cigna	Silver	3	Colorado3	158.09	248.96	248.96	248.96	248.96	249.96	254.93	260.91	270.62	278.59	282.57	288.54	294.52	298.25	302.24	304.23	306.22
7	Colorado Choice	Silver	2	Colorado2	131.21	206.64	206.64	206.64	206.64	207.46	211.59	216.55	224.61	231.22	234.53	239.49	244.45	247.55	250.86	252.51	254.16
8	Colorado Choice	Silver	3	Colorado3	146.29	230.38	230.38	230.38	230.38	231.30	235.91	241.44	250.43	257.80	261.49	267.01	272.54	276.00	279.69	281.53	283.37
9	Colorado Choice	Silver	4	Colorado4	177.97	280.27	280.27	280.27	280.27	281.39	286.99	293.72	304.65	313.62	318.10	324.63	331.55	335.76	340.24	342.49	344.73
10	Colorado Choice	Silver	6	Colorado6	180.98	285.02	285.02	285.02	285.02	286.16	291.86	298.70	309.81	318.93	323.49	330.33	337.17	341.45	346.01	348.29	350.57
11	Colorado Choice	Silver	8	Colorado8	150.82	237.51	237.51	237.51	237.51	238.46	243.21	248.91	258.18	265.78	269.58	275.26	280.98	284.54	288.34	290.24	292.14
12	Colorado Choice	Silver	9	Colorado9	177.97	280.27	280.27	280.27	280.27	281.39	286.99	293.72	304.65	313.62	318.10	324.63	331.55	335.76	340.24	342.49	344.73
13	COOP	Silver	1	Colorado1	132.15	208.11	208.11	208.11	208.11	208.94	213.10	218.10	226.21	232.87	236.20	241.20	246.19	249.31	252.64	254.31	255.97
14	COOP	Silver	3	Colorado3	135.59	213.53	213.53	213.53	213.53	214.38	218.65	223.78	232.11	238.94	242.36	247.48	252.60	255.81	259.22	260.93	262.64
15	COOP	Silver	4	Colorado4	148.90	234.49	234.49	234.49	234.49	235.43	240.12	245.75	254.89	262.40	266.15	271.76	277.40	280.92	284.67	286.55	288.42
16	COOP	Silver	6	Colorado6	143.94	226.67	226.67	226.67	226.67	227.58	232.11	237.55	246.40	253.65	257.28	262.72	268.16	271.56	275.18	277.00	278.81
17	COOP	Silver	9	Colorado9	177.42	279.40	279.40	279.40	279.40	280.52	286.10	292.81	303.71	312.65	317.12	323.82	330.53	334.72	339.19	341.42	343.66
18	COOP	Silver	10	Colorado10	190.84	300.53	300.53	300.53	300.53	301.73	307.74	314.96	326.68	336.30	341.10	348.32	355.53	360.04	364.85	367.25	369.65
19	COOP	Silver	11	Colorado11	229.50	361.42	361.42	361.42	361.42	362.87	370.10	378.77	392.67	404.43	410.22	418.89	427.56	432.99	438.77	441.66	444.55
20	COOP	Silver	1	Colorado1	152.09	239.51	239.51	239.51	239.51	240.47	245.26	251.01	260.35	268.02	271.85	277.60	283.35	286.94	290.77	292.69	294.60
21	COOP	Silver	2	Colorado2	153.28	241.39	241.39	241.39	241.39	242.36	247.19	252.98	262.39	270.12	273.98	279.77	285.57	289.19	293.05	294.98	296.91
22	COOP	Silver	3	Colorado3	156.05	245.75	245.75	245.75	245.75	246.73	251.65	257.55	267.13	275.00	278.93	284.83	290.72	294.41	298.34	300.31	302.27
23	COOP	Silver	4	Colorado4	171.35	269.84	269.84	269.84	269.84	270.92	276.32	282.80	293.32	301.95	306.27	312.75	319.22	323.27	327.59	329.75	331.91
24	COOP	Silver	5	Colorado5	202.64	319.12	319.12	319.12	319.12	320.39	326.77	334.43	346.88	357.09	362.20	369.86	377.51	382.30	387.41	389.96	392.51
25	COOP	Silver	6	Colorado6	165.05	260.86	260.86	260.86	260.86	261.90	267.12	273.38	283.56	291.90	296.08	302.34	308.60	312.51	316.68	318.77	320.86
26	COOP	Silver	7	Colorado7	167.91	264.43	264.43	264.43	264.43	265.49	270.78	277.12	287.43	295.90	300.13	306.47	312.82	316.79	321.02	323.13	325.25
27	COOP	Silver	8	Colorado8	160.67	253.03	253.03	253.03	253.03	254.04	259.10	265.17	275.04	283.14	287.19	293.26	299.33	303.13	307.18	309.20	311.22
28	COOP	Silver	9	Colorado9	204.80	322.53	322.53	322.53	322.53	323.82	330.27	338.01	350.59	360.91	366.07	373.81	381.55	386.39	391.55	394.13	396.71
29	COOP	Silver	10	Colorado10	220.68	347.54	347.54	347.54	347.54	348.93	355.88	364.22	377.77	388.89	394.45	402.79	411.13	416.35	421.91	424.69	427.47
30	COOP	Silver	11	Colorado11	265.16	417.58	417.58	417.58	417.58	419.25	427.60	437.63	453.91	467.27	473.96	483.98	494.00	500.26	506.94	510.29	513.63
31	COOP	Silver	3	Colorado3	158.46	249.54	249.54	249.54	249.54	250.54	255.53	261.52	271.25	279.23	283.23	289.21	295.20	298.95	302.94	304.94	306.93
32	Denver Health	Silver	3	Colorado3	136.55	215.05	215.05	215.05	215.05	215.91	220.21	225.37	233.76	240.64	244.08	249.24	254.40	257.63	261.07	262.79	264.51
33	Denver Health	Silver	3	Colorado3	136.55	215.05	215.05	215.05	215.05	215.91	220.21	225.37	233.76	240.64	244.08	249.24	254.40	257.63	261.07	262.79	264.51

To analyze this data – and present it in a visualization – I want each age to have its own row. So instead of one row for each insurance project in a rating area, we'll end up with 45 (there are 45 age groups in this data)

To reshape, put your cursor on the first data point – in this case F2 – and launch the reshapener. It will push the data out to a new worksheet.

Note: If the new data file exceeds 1 million rows, this will automatically export your results as a .CSV file