

MaryJo (Sylwester) Webster
St. Paul Pioneer Press
mwebster@pioneerpress.com
Revised: March 2007

#2799
13pg

My Favorite (Excel) Things

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 Tricks".

Date Functions:

Month-Day-Year:

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)

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)

Displaying data as a 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:

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)

Other text functions:

EXACT(text1, text2): 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): 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:

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	VisitAbbr	HomeAbbr	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:

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:

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 example we're going to use is very straightforward. It just has three IF statements. We're going to use this to assign our data to categories. The data lists results of regression analyses for St. Paul public schools in 3rd grade math. When you run a regression it assigns a predicted score, then compares that to the school's actual average score. The difference between those two numbers is called a "residual."

SCH NAME	No. Students	LONG TERM	POVERTY	NO SPEC	predicted	actual	residual	Category
BATTLE CREEK MAGNET EL	1548	67.5%	51.8%	11.7%	1435	113	2.46	
ST. ANTHONY PARK EL	1654	24.6%	15.0%	8.8%	1555	99	2.15	
ROOSEVELT MAGNET EL	1481	90.0%	54.7%	7.6%	1386	96	2.07	
PARKWAY EL	1490	89.0%	46.2%	13.9%	1395	95	2.06	
EXPO/HARRIET BISHOP CENTER	1589	46.5%	26.0%	15.2%	1498	91	1.98	
CAPITOL HILL MAGNET/RONDO	1624	30.4%	21.7%	4.5%	1537	87	1.89	
FARNSWORTH EL	1500	77.0%	52.1%	10.5%	1415	85	1.85	
CHELSEA HEIGHTS EL	1587	45.4%	20.0%	12.2%	1507	80	1.73	
MANN EL	1630	22.5%	12.0%	17.2%	1562	68	1.49	
SHERIDAN EL	1507	71.7%	33.9%	11.7%	1441	66	1.44	
FRANKLIN MAGNET EL	1430	89.1%	62.8%	8.1%	1381	49	1.06	
MISSISSIPPI MAGNET EL	1442	83.5%	60.8%	20.5%	1395	47	1.03	
PROSPERITY HEIGHTS EL	1463	79.2%	43.9%	15.4%	1417	45	0.98	
RIVERVIEW MAGNET EL	1442	84.2%	50.0%	11.7%	1402	40	0.88	
MUSEUM MAGNET/RONDO	1507	60.5%	31.8%	10.8%	1466	40	0.88	

Whether a school performed better than expected, after adjusting for factors like poverty, is determined by the standard deviation of your residuals. In the example here, the SPSS program told me the standard deviation was 46. In other words, any schools with a residual above 46 could

be deemed “above predicted.” Those below negative 46 would be “below predicted” and everything else in between is considered within predicted.

In this example, the order that we assign the IF statements doesn’t really matter. But oftentimes with nested IF statements you need to consider the order. Here’s why.

If we wanted just 1 IF statement to assign the phrase “above predicted” to a new field and to leave it blank otherwise, this is what it would look like:

=IF(G4>46, “above predicted”, “”)

To nest an IF statement, we’re going to put the 2nd one in the place of the FALSE portion of the first IF statement. In other words, it will only go on to the 2nd statement if the first statement is false. Here’s what it would look like if we put those 2 together.

=IF(G4>46, “above predicted”, IF(AND(G4<47, G4>-47), “within predicted”, “Do nothing”))

This is how it is structured:

=IF(CRITERIA1, TRUE1, IF(CRITERIA2, TRUE2, FALSE2))

Notice that you don’t see the false portion of the first IF statement. It’s actually comprised of the entire 2nd IF statement. So here’s what it’s doing: If G4>46, it’s going to put the words “above predicted” in our column. If it’s false, it will determine whether G4 falls between 47 and -47 and if that’s true, it will put “within predicted” in our column. If it’s false, it will put “Do nothing” in our column.

A quick note about the AND: This is another logical function available in Excel. It allows you to set two criteria within that one IF statement. So in this case we’re saying G4 must be less than 47 and it must be greater than -47. There is also an OR function that works the same way, except obviously, that it will provide a broader sweep.

So now to add the 3rd IF statement, we’re going to put that in the spot where we have “Do nothing”. Be sure that your last IF statement includes a false portion (in this example it’s two double quotes, indicating we want to leave the cell blank)

Here’s the final formula:

=IF(G4>46, “above predicted”, IF(AND(G4<47, G4>-47), “within predicted”, IF(G4<-46, “below predicted”, “”)))

SUMIF and COUNTIF Functions:

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!e2:Salaries!e424)`

	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(N3:N90, ">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.

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 hundreth (1200)
- -3 to the nearest thousandth. (1000)

Lookup Tables:

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:

	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
9	MN	27	013	Blue 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

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. 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.

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.**

Custom lists:

If you've got a list that you use frequently in Excel — such as state names or city names — you can create a custom list that will allow you to use the copy-down function in Excel to insert those

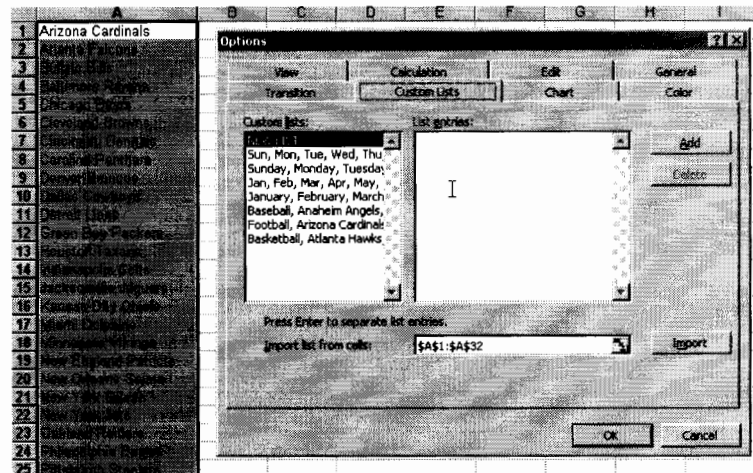
names into a new worksheet. (This is why you can type a day of the week or month and get it to change as you copy down or across)

To create a custom list from an existing list in Excel:

- 1) Highlight the list.
- 2) Go to the Tools Menu and choose Options. Navigate to the "Custom Lists" tab.

You should see the range of your data in the "import list from cells" (the days of the week and months are defaults)

You can also type the list directly into the "list entries" box and push the "Add" button.



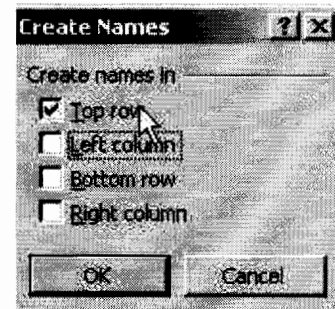
To recall the list and insert into a new worksheet:

Simply type the first item in your list, then use the copy-down tool.

Hint: For my lists of sports teams (see picture) I decided to make the first item in the list the sport, such as baseball, football and basketball, so that I merely have to type the sport and I will get the list. I don't have to remember which team is the first one in the list.

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). 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.



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.

File Edit View Insert Format Tools Data Window				
Arial 10 B I U				
D2		=WEEKDAY(DOB)		
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	Bouman	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.