Get more from your Web Management Reports

Customizing PTYPE and location checkout reports using Excel

Presented by
John Sterbenz
Associate Librarian
Manager, Technical Services and Library Automation
Kresge Business Administration Library
Ross School of Business
The University of Michigan

(734) 764-5746
jsterben@bus.umich.edu

EGL-IUG 2009 Meeting
Akron-Summit County Public Library
September 25, 2009

Introduction

With proper data collection and a few additional minutes of time, Excel provides us with ways of summarizing Web Management Report data in ways far beyond what the INNOPAC can provide us. This hands-on session hopes to impart some of this knowledge onto you while introducing you to some Excel functionality you might not have known existed.

The overarching process for both types of reports we'll be examining today (location and PTYPE) is as follows:

1) Gather data using the Web Management Reports' Spreadsheet Interface
2) Create "Master lists" of location codes and PTYPES, amending all entries in these lists with fields that we're interested in summarizing our data by
3) Create a comprehensive table of checkout activity using the lists created in step 2)
4) Use Excel's PivotTable feature to summarize the tables created in step 3)
5) Customize PivotTable displays (optional)

Throughout this workbook:
Actions you need to perform are typed in *italics*
Excel functions and formulas you need to type will be in `Courier font`

LOCATION-BASED REPORTS

1. Gather data using Web Management Reports' Spreadsheet Interface

We can generate a variety of circulation reports using Millennium's Web Management Reports interface. The URL to access these reports is http://<your INNOPAC IP address or machine name>/iii/webrpt, *e.g.*

http://lib.bus.umich.edu/iii/webrpt/, or
http://141.211.86.45/iii/webrpt/

Reports generated using this interface (called the "Spreadsheet Interface" by III) will be the subject of our scrutiny today. The Spreadsheet Interface is available in Release 2007 and beyond. The traditional interface (also called the "Web Browser Interface") available at
http://<IP address or machine name>/manage remains available to Release 2007 and 2009 users and is the only version of Web Management Reports available for versions of the INNOPAC software earlier than Release 2007.

We'll focus on Spreadsheet Interface reports because, unlike any report generated using the Web Browser Interface, Spreadsheet Interface reports are provided to us as Excel files. This saves us the process of having to download tab-delimited files for importing into Excel required with the Web Browser Interface. With minor modifications, what we're doing with the Spreadsheet Interface can be adapted to the other interface, however.

Millennium Guide and Reference #107508 describes the different reports currently available in the Spreadsheet Interface. You must be authorized for function 017 (Circulation Statistics) to generate the reports required for this session. To review (but not create) reports created in this interface, users must be authorized for function 677 (View Web Management Reports 2007).

Open Web Management Reports' Spreadsheet Interface
(http://<IP_address_or_machine_name>/iii/webrpt/).

Make sure "CREATE" tab is in focus.

Click on "All Circulation Activity".

Provide an authorized set of credentials ("initials and password") when prompted.

Set the date range as desired using the "Dates" dropdown.
- 36 months of data are available, including the current month. Since we might be interested in constructing a month-to-month comparison, either with any given year or between years, data collection for this session has been performed at the monthly level using the "User-specified" option.

Set "Sort By" to "Item Location".

Click on "Submit".

When the dialog box appears asking what to do with the report, select "Open with" and choose "Microsoft Office Excel", which should be the default. Click on "OK".

Open "200807AllCirc.xlsx" (a typical report, created for a single month)

There are several worksheets included within the report. We'll work with the "Raw Data" worksheet. Other worksheets may rely on cell referencing to pull values; all data in any "Raw Data" worksheet is flat text and thus most suitable for "cut and paste" operations between workbooks. Though I've combined the "Raw Data" worksheet from each month's workbook in a new workbook for the purposes of this session, it is also possible to work with the individual Excel files created for each month, referencing the "Raw Data" worksheet within each individual workbook.

Observe that only locations with activity are reported on any given worksheet. These locations are going to vary from month to month. The location listed in row 20 in July might be listed in row 25 in August and not listed at all in September—a condition particularly true for INN-Reach locations.

We need to somehow account for these different row locations. Here's how: We'll create a "master list" of all location codes and their corresponding long names, creating new columns for every month of data we want to capture. While we're at it, we'll also add columns for each additional characteristic of the location codes we want to measure and summarize by, like physical branch and subcollection. Then, we'll use VLOOKUP to populate the master list "month" columns with circulation numbers, pulling data from the WMR-generated reports and run analysis on that.

2. Create "Master list" of location codes, amending all entries in these lists with fields that we're interested in summarizing our data with

The master list of location codes and their corresponding long names is available only in the character-based system (Millennium will only list the codes), from the command sequence:

 M (MANAGEMENT information)
  I (INFORMATION about the system)
    C (CODES used)
      B (BRANCH codes)

Printed to the e-mail printer, this master list can be saved as a text file and imported into Excel as a fixed-width file for easy parsing of the file into codes and names. Let's do this.

In Excel, open the file "Circ Branches.txt".

- This file is the body of the e-mail, in its entirety, that the INNOPAC will generate when sending the locations list to the e-mail printer.

Text Import Wizard Step 1: Choose "Fixed Width" radio button. Start import at Row 5. Click "Next".

Text Import Wizard Step 2: Create line breaks at position 6 (just before location code) and 16 (just before location long name). Click "Next".

Text Import Wizard Step 3: Do not import first column (line numbers). Import columns 2 and 3 as "Text". Click "Finish".

With our location codes and their corresponding long names in separate columns, we'll now add additional columns to reflect characteristics we are interested in summarizing our checkout data by. In column C, we'll list physical branch. In column D, we'll include subcollection. In your data, include as many different characteristics as you like, each in their own column. Some things to note:
- Labels within a single column must be mutually exclusive. That is to say that any given location cannot fit in more than one label description for any given column.
- Text entry between labels in the same column must be exact (including spacing) for Excel's PivotTable functionality to properly combine them. Excel's AutoComplete feature can help to ensure this requirement is met.
- Different columns do not need to be subsets of other columns as they are in this example. Each column will be able to function independently of other columns when used by the PivotTable.
- Not all columns need to be defined for all locations. However, when leaving any given entry in a column blank, a single space should be placed in the cell to prevent Excel from displaying an error when VLOOKUP is used.

The end result of this necessary preparatory work can be found in the file "Circ Branches.xlsx", which has a header row added to it and has all labels provided for the two fields we'll want to summarize by (Branch and Subcollection) .

Open "Circ Branches.xlsx".
- Circ Branches.xlsx serves as the "Master list" of location codes we want to work with for this example.

3. Create a comprehensive table of checkout activity using the lists created in step 2

Let's complete our header row in "Circ Branches.xlsx".

For the other cells in row 1 of "Circ Branches", use a month-year approach to label creation. In our instance, label cells E1 through P1 "July 2008" to "June 2009". Use the fill handle after completing E1 to make the job easy!

Open the file "FY09AllCirc.xlsx".
- "FY09AllCirc.xlsx" contains 12 months of "Raw Data" worksheets from Excel workbooks created with the Spreadsheet Interface. To ease entry of the external workbook reference we're about to make, we want this file open.

Now, we'll use VLOOKUP to pull in the checkout numbers for all of the different location codes.

Back in "Circ Branches.xlsx", in E2, enter the formula

```
=VLOOKUP($A2,'[FY09AllCirc.xlsx]200807'!$A:$H,2,FALSE)
```

- Use point-and-click methods to easily generate the table_array (second) argument of VLOOKUP.
- We want to keep all column references absolute (prefaced with a dollar sign) but will keep all row references relative (no dollar sign)
- **If you prefer, open the file "function calls.txt" in Notepad and copy/paste Line 1 into E2.**

Not quite what we expected, is it? Our VLOOKUP function is working fine….the problem lies in the fact that this particular location code (admin) is "Not available" in the table array being referenced, so Excel returns the #N/A error.

We need to account for the error using the Excel ISERROR function. If the VLOOKUP function call returns any error at all (of which #N/A is one), we should have it display nothing—otherwise, it should return the value. PivotTables do not respond nicely to the presence of errors in fields that end up in the "Values" area, so embedding our VLOOKUP function within an ISERROR function is necessary.

Read the above again, more carefully. We've outlined exactly what we need to do. At a higher level, our function call would read like this:

=IF(ISERROR([VLOOKUP function call]),"",[VLOOKUP function call])

While we *could* change the function call in E2, we're instead going to place the function call in a cell I know will yield a real numerical result: E28

In E28 of "Circ Branches.xlsx", enter the following formula:

```
=IF(ISERROR(VLOOKUP($A28,'[FY09AllCirc.xlsx]200807'!$A:$H,2,FALSE)),
"", VLOOKUP($A28,'[FY09AllCirc.xlsx]200807'!$A:$H,2,FALSE))
```

Use point-and-click for table array calls at your discretion.

- **Or, copy Line 2 from "function calls.txt" and paste it in E28.**

If you've typed and point-and-clicked (or even just copy/pasted!) everything correctly into E28, the formula returns a value of 259.

The use of absolute references for columns will allow us to use the fill hand across the range of cells F28:P28. We then only need to change the month in the worksheet references of the table arrays in *both* VLOOKUP function calls.

Select E28 and use the fill handle to populate the range of cells F28:P28.

All values in row 28 read "259". Of course they do, since all of the formulas point to the same data from July 2008. Now we need to manually (unfortunately!) change the month/year references in each of the remaining cells.

Change the year/month numbers in the table_array arguments for the rest of row 28, e.g. for F28 (August 2008), any reference to "200807" should be changed to "200808".
- As previously mentioned, there will be two such changes that will need to be made in every cell. Don't forget to change the year in January (and for all six months in 2009!)

If you've done everything correctly, row 28 will look like this:

26	rsvv	Circ. Desk. Videos (2nd Floor)	Kresge	Reserves											
27	rsvz	Circ. Desk. Reserves (2nd Floor)	Kresge	Reserves											
28	sta	Stacks (4th Floor)	Kresge	Stacks	259	250	488	491	336	305	531	371	473	378	305	274
29	staa	Art on View (2nd Floor)	Kresge	Stacks											

If you choose to keep each month in its own workbook, not that if Excel prompts you for a file location after you hit Enter to finalize any given formula, it is telling you it can't find the file you specified, and is asking you to select one so it can create the proper link. Recheck your typing or navigate and select the correct file. Be sure to also check that your date references for workbooks are correct.

Entering the initial formula, copy/pasting, and then changing all of the file references for the other months is EASILY the most difficult portion of this job (and of this session). But your few minutes of effort and diligence is about to pay off, for with one record completed (the one for "stacks"), we can now use the fill handle to complete every other record in our list.

Select the range of cells E28:P28. Use the fill handle to populate the rest of the table in one direction (down or up), waiting for Excel to finish all of its work before going in the opposite direction. Watch the status bar to determine progress. This can take awhile if the workbook being referenced is closed!

It is not yet time to be impressed. All we've really done at this point is pull in all of our data points from across 12 different worksheets from another workbook into a single worksheet so we can perform analysis on them. Well, we can be a little impressed and a lot relieved that we just made Excel do all of the dirty work for us.

Let's create one additional column that adds up all of the activity for any given location for the entire fiscal year:

In cell Q1, type a label such as "FY 2009"

In cell Q2, type:

```
=SUM(E2:P2)
```

Use the fill handle to populate the FY 2009 sums for the remaining 441 locations in column Q.

4. Use Excel's PivotTable feature to summarize the table created in step 3

With our "Master List" of locations now populated with checkout values, we'll create a PivotTable, placing "Branch", then "Subcoll" in the "Rows" area and each of the months in the "Values" area.

If needed, open the workbook "Circ Branches with data.xlsx".

- If you happened to get lost or made a mistake along the way, opening "Circ Branches with data.xlsx" gets you back to where you need to be. Though all references in the following instruction will list "Circ Branches.xlsx", the same steps can be performed in "Circ Branches with data.xlsx".

Make sure your active cell is somewhere in your list of circulation data in "Circ Branches.xlsx".

Open the "Insert" ribbon. In the "Tables" subgroup, click on the "PivotTable" icon.

In the "Create PivotTable" dialog box, be sure the range of cells is set to $A$1 to $Q$443 and place the PivotTable in a new worksheet.

With the PivotTable framework established:

Place "Branch", then "Subcollection" in the "Rows" area of the PivotTable

Add each of the months and the FY Total fields to the "Data" area of the PivotTable.

Since the data fields contain blank cells, Excel incorrectly thinks we want counts of non-empty cells and not sums. Let's correct this.

Right-click in any "Date" column in the "Data" area. From the context menu, select "Summarize Data By", then "Sum". Repeat for all months and the FY 2009 totals.

NOW we can be impressed. Millennium, while it **does** give us breakdowns by individual location codes, **doesn't** do anything to group them for us in a meaningful manner. By identifying each location code with a branch (and a subcollection within that branch, if we wanted, so we could compare, say, "stacks" collections across branches), we can use the power of PivotTables to generate these numbers quickly and easily. We could use this power across other characteristics as well had we provided additional columns of labels in "Circ Branches.xlsx".

Interested in viewing this by "Subcoll", then "Branch"? Simply switch their order in the "Row" area of the "Field List".
- Given the particulars of this dataset, this is not as pretty a display because many subcollections are associated with only a single branch, namely MeLCat.

Interested in the breakdown by actual item location, not subcollection? Remove "Subcollection" from the "Row"area. Drag "Loc Code" or "Long Name" from the "Field List" to the "Row" area, placing it underneath "Branches".
- This view rearranges the data underlying the PivotTable in such a way that all location codes affiliated with any given branch are efficiently collected.

In Web Management Report's traditional (Web Browser) interface, we can get a report *close* to this (without additional transcription). To get exactly what we see here, though, several additional steps are required, and at least one of them involves transcription. Any time transcription is involved, we introduce the possibility of error. In our exercise, consider how many data points we entered manually. The answer is NONE. All of our data points (12 months * 442 locations = 5,304) were gathered by using VLOOKUP. How many calculations did we perform? Again, the answer is NONE. We used the power of PivotTables in Excel to do all of the work for us. Creating additional fields by which to summarize our data (branches and subcollections) allowed us to add a new dimension to our reporting as well. While we'd have been able to gather a report branch by branch, there is no way to do what we did. The secondary summation we did by subcollection is not possible within Millennium. Nor is it possible to so easily switch views to make subcollection the primary focus instead of branches. The PivotTable made this as easy as changing a single option!

5. Customize PivotTable displays (optional)

Once you have all of the row, column, and data areas set, consider customizing the PivotTable display. Excel 2007 introduced many new PivotTable customization features. We'll discuss ones relevant to the tables we've created here. Most of the customization options are available through the PivotTable tool sets that will appear as different colored tabs above the ribbon while working within the PivotTable and in right-click context menus that will appear when the cell pointer is placed in the appropriate position in the PivotTable.

Displaying/hiding field headers (this makes your report more professional looking)
- Place the active cell in the PivotTable to display the PivotTable tool sets
- Go to the "Options" ribbon
- In the "Show/Hide" group, click on "Field headers"

Displaying/hiding expand/collapse buttons
- Place the active cell in the PivotTable to display the PivotTable tool sets
- Go to the "Options" ribbon
- In the "Show/Hide" group, click on "Buttons"

Expanding/collapsing Row/Column view
- Display expand/collapse buttons
- Click on the buttons immediately left of any given label

Removing/adding individual labels from Row/Column areas
- Display field headers
- Click on the drop-down arrow next to "Row Labels"
- Select the appropriate field from the "Select Field" dropdown
- Clear or select checkboxes as appropriate from the label listing near the bottom of the dropdown
  - If you select a label that appears in more than one hierarchical level above (such as "Stacks" from "Subcollection"), the selected label will display/not display for all of them

Moving labels in Row/Column areas
- Right-click on the label you want to move to bring up the context menu
- Select the "Move" option
- Choose from the menu: Move "[selected label]" to Beginning, Up, Down, to End
  - If you select a label that appears in more than one hierarchical level above, the selected label will switch positions in all of the higher levels, not just the selected level.

Closing/displaying field list task pane
- Right-click anywhere in the PivotTable to bring up a context menu
- Select final option: Show Field List/Hide Field List

Changing column header labels
- Right-click anywhere in the column whose display header you want to change to display the context menu
- Select "Value Field Settings"
- Change label in "Custom Name" text box.
  - You cannot use pre-existing field names

Customizing PivotTable display options
- Place the active cell in the PivotTable to display the PivotTable tool sets
- Go to the "Design" ribbon
- Choose an option from the "PivotTable Styles" group. You can also create your own!

No matter what customizations you choose, always remember that altering display of the PivotTable or its various elements affect only the PivotTable itself. The underlying data is never modified or deleted as a result of any action taken against a PivotTable.

Removing zero-totalled rows

We could remove zero-totalled rows individually by following the steps outlined in "Removing/adding labels from Row/Column areas" discussed above. There's a better way to do this, however. We'll run our Master List through Excel's "Advanced Filter", placing rows that display through the filter in a new part of the worksheet, creating a PivotTable from that.

Why go through all this? When a PivotTable is invoked, all records (rows) of a list are included within the table, even if suppressed by the invocation of a filter. Advanced Filter, with its ability to extract filtered records to another location, becomes a necessity.

Return to the "Master list" of either "Circ Branches.xlsx" ("Branches" worksheet) or "Circ Branches with data.xlsx" (Sheet1 worksheet).

Insert 5 blank rows at the top of the Master List.

In A1, enter the header row label for column Q, e.g. FY 2009
- Case does not matter, but spacing does!

In A2, enter the following conditional test:
>0

Place the active cell in the Master List.

Go to the "Data" ribbon. In the "Sort & Filter" group, click on "Advanced".

In the "Advanced Filter" dialog box:
- *Select radio button "Copy to another location"*
- *List range: $A$6:$Q$448*
- *Criteria range: $A$1:$A$2*
Copy to: $A$450
Click "OK".

Place the active cell in the newly-filtered data.

Create a PivotTable for the range of cells A450:Q587
- *Branch, then Subcollection go in Rows*
- *Months, FY 2009 fields go in Data*

Change counts to sums for columns using the right-click context menus.

Voila!

Some parting observations

We could create similar reports for any activity captured with the "All Activity" report generated at the start of this endeavor simply by changing the columns referenced in the VLOOKUP function. We could do this for any type of report that lists its data in column form. Here are the key things to remember:

- You must account for any possible value/label in the left-most column of the Millennium-generated report. Buried within character-based "MANAGEMENT Information → INFORMATION about the system → CODES used" menu lie all such tables, which are generally more complete than the information that would display via any of the "Admin" menu options in any given Millennium module.
- You must account for location codes that may have been in the system at the time the data was collected but have since been removed. Failure to do this will cause discrepancies between the Web Management Reports values and those calculated through the PivotTables.
- All data being summarized can be placed in a table using VLOOKUP functions that can pull these data points in from another source (be they external files or other worksheets in the same workbook).
- You must account for data not available when VLOOKUP references any given table by using the ISERROR function, placing it within an IF function call using the following natural language syntax: "IF VLOOKUP returns an error, keep the cell blank. Otherwise, provide me with the value VLOOKUP finds".
- When creating your own subcategories, labels must be identical between codes you wish to combine. This includes use of trailing spaces, as necessary. You must also be mindful of the relationship between the data being analyzed and the labels used to translate them using VLOOKUP.

Close all files except function calls.txt. It is not necessary to save changes.

PTYPE REPORTS

In our previous exercise, I noted that it was possible for Millennium to provide us with a report based on branches. This is possible because, in our system configuration, we can use the "Locations Served" table to group various location codes together and can create reports in that way.

We are not given this same flexibility with PTYPES, though—we can't group them together in the system. Many of us likely have implicit higher-level groupings of PTYPES that we use, with these higher-level groupings (such as branch or campus affiliations) usually reflected in the manner in which we'll group various PTYPES together when creating them.

Using the same tools we used in the last exercise, but on a different report from Web Management Statistics, we can create more meaningful reports on circulation statistics based on PTYPE. Because the methodologies are nearly identical, much of the process here is glossed over. Refer to the "locations" examples for complete explanations and details.

The "All Activity" report generated for our last exercise cannot be sorted by PTYPE. We'll gather some new checkout reports.

(Re-)open up Web Management Reports' Spreadsheet Interface.

Display the "CREATE" tab.

Click on "Checkouts".

Provide an authorized set of credentials ("initials and password") if prompted.

Under "Dates", select "User spec" and set (we'll again capture monthly data). Under "Locations", select "All Locations". Under "Report by", select "PTYPE".
- As with all circulation reports, 36 months of data are available, including the current month.

Click on "Submit".

Generate all such reports and save the resulting Excel workbooks.

Open "200807PTYPECirc.xlsx".

This is a typical monthly file. Again, because it contains flat text, we'll work with the "Raw Data" worksheet. While we could work with the "Raw Data" worksheets from all of these files, we'll instead consolidate all of the "Raw Data" worksheets into a single workbook just as we did with our previous example.

Open "FY09PTYPECirc.xlsx"
- This is the resultant consolidated workbook

From the character-based system, generate a list of all possible PTYPE values (from the Main Menu, M→I→C→X). This should be sent to the e-mail printer and imported into Excel. Fields other than PTYPE can be stripped either prior to importing in Excel or after the file has been converted. When converting numeric PTYPE values that have leading zeros, these MUST be imported as text to preserve them. Though this particular example relies on the use of long names and not the numeric values, we will want to keep both in our table that will serve as our translation table.

Open "PTYPES.xlsx". The file has already been prepared for you. Note the addition of two columns that we will use to combine similar PTYPES for reporting purposes—one based on affiliation, the other on level.

In cell E1, type "July 2008". Hit Enter, make E1 the active cell, and use the fill handle on E1 to enter the rest of the fiscal year months across the range of cells F1:P1.

Invoke the data call for the PTYPE/date combination in cell E2, remembering to account for the #N/A error that could be returned by VLOOKUP by using a combination of the ISERROR and IF functions. Because "PTYPE checkouts 200607.xls" is open, the function call for July will look like this:

```
=IF(ISERROR(VLOOKUP($B2,'[FY09PTYPECirc.xlsx]200807'!$A:$C,3,FALSE)),"
",VLOOKUP($B2,'[FY09PTYPECirc.xlsx]200807'!$A:$C,3,FALSE))
```

If you prefer, refer to Line 3 of "function calls.txt" and copy/paste Line 3 into E2.

We use B2 as the VLOOKUP lookup_val because the long name appears in our Millennium-generated reports. We make the column reference absolute to allow us the opportunity to use the fill handle across the rest of the row.

With the July 2008 value pulled in (noting it is actually a blank given our formula), make E2 the active cell and grab its fill handle, using that across the range of cells F2:P2.

Select each cell of F2 to P2 individually. In the formula bar of the active cell, change all yyyymm references in worksheet names to the month listed in row 1 for all of the formulas. There will be two changes to make in each formula.

Provided all mmmmyy changes have been made correctly, row 2 should now look like this:

	K9		f_x													
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	PTYPE	PTYPE Label	Affiliation	Level	Jul-08	Aug-08	Sep-08	Oct-08	Nov-08	Dec-08	Jan-09	Feb-09	Mar-09	Apr-09	May-09	Jun-09
2	000	INVALID: SELECT NON-ZERO VALUE	ERROR	ERROR			2	2	2	4	6	1				
3	001	KBAL Librarian	Ross School	Faculty												

Once all yyyymm changes have been made, highlight the range of cells E2:P2 to select all twelve monthly formulas. Use the fill handle to complete calculations for all remaining PTYPES.

Open "PTYPES with data.xlsx".

All data is made available to you in this workbook without your having to make all of the date changes or populate other records. This workbook also has the FY summary column added to it.

Create the PivotTable on a new worksheet, ensuring the range of cells is A1:Q56.
- "Affiliation" and/or "Level" go into the "Rows" area.
- All months and the FY summary go into "Values".
- Note that all months will need to be changed from "Count" to "Sum" via "Value Field Settings" or the context menu "Summarize Data By…" option.

Set your desired design in the context-specific Design tab.

Change the order of display of affiliation to list Ross first by right-clicking on the "Ross School" affiliation, selecting "Move", then choosing "Move 'Ross School' to Beginning". Selecting a "Level" will move that (across all higher levels) instead of moving the desired affiliation.

Close all files. It is not necessary to save changes.

Excel function reference

Here's the nitty-gritty on the more complicated functions we used in the session:

IF

Format: `IF(logical_test, value_if_true, value_if_false)`
where:
`logical_test` returns either `TRUE` or `FALSE`. The other arguments display based on the results of `logical_test`

VLOOKUP

Format: `VLOOKUP(lookup_val, table_array, col_index_num, range_lookup)`,
where:
`lookup_val` is the value, cell reference, or text (in quotes) that you want to find in a table or range of cells
`table_array` is a cell range that defines the table in which to look
`col_index_num` is the column *number* of the table from which to select the result, counted *relative to the table*
`range_lookup` is a Boolean (`TRUE` or `FALSE`) that determines whether the function matches `lookup_val` exactly or approximately. Use `TRUE` for approximation; use `FALSE` to force exact matching.

ISERROR

Format: `ISERROR(value)`

`ISERROR` will return only `TRUE` or `FALSE`. If `value`, when fully evaluated will return any error (including `#N/A`), `ISERROR` returns `TRUE`. If `value`, when fully evaluated does not return an error, `ISERROR` returns `FALSE`.

IFERROR

Format: `IFERROR(value, value_if_error)`

This replaces the more cumbersome IF(ISERROR(VLOOKUP(…)),""),VLOOKUP(…)) notation used in this session. For a point of example, consider the following from our "locations" exercise. Essentially, `IFERROR` works as follows: If `value` produces a value, display it as the result of `IFERROR`. If `value` produces an error, display the second argument of `IFERROR`. `IFERROR` is new in Excel 2007. If only I'd discovered it before Tuesday, September 22!

`=IFERROR(VLOOKUP($A28,'[FY09AllCirc.xlsx]200807'!$A:$H,3,FALSE),"")`