ICPI Tool Style Guide

A. Chafetz, USAID

April 13, 2016

OVERVIEW

Over the last year, PEPFAR has been setting targets and entering into DATIM since early last year. With a wealth of HIV/AIDS programmatic results down to the site level quarterly, DATIM is an extremely useful resource to support data-driven decision making and help build capacity necessary towards achieving epidemic control. DATIM has features, such as its built in pivot table and visualization that helps get a view of the data. This view, however, is just a peak into the data and can be rather limiting.

Agencies, TWGs, and ICPI workstreams need to get more than a glimpse at this data; they need to use this information to evaluate progress and make strategic decisions. As a result, we are seeing an influx in "tools" and "dashboards" to make sense of complex and ample data across numerous operating units and indicators. This is a great step towards data-driven decision making. One drawback is the lack of coordination and, as a result, a lack of uniformity in the creation of these tools.

An optimal solution would be to coordinate efforts, designing visual tools and dashboards in a similar fashion. Adopting a uniform structure and overall style has a number of benefits including a similar SQL pull for each product to get large scale data out of DATIM and improved collaboration across ICPI analysts.

Given the widespread use of Excel given accessibility and skillsets, this document outlines style and structure guidance for developing dashboards and tools in Excel.

STARTING WITH PEN AND PAPER

The first step even before touching any data is to have a plan of attack. The best way to do this is to start with pen and paper. Begin by jotting down the purpose of the visualization/tool and the message you hope to get across to the audience. Think about the ways in which you can visualize this and how different tables and figures can reinforce this or multiple points. If you are making a dynamic tool, consider how you want the user to interact with the interface and adjust the views and data.

With your concept in mind, you can begin to think about the design of your tool by sketching out some rough ideas of the structure. You do not need to be an artist to do this; its more the process of thinking about how the pieces will be placed, designing the overall layout, and how different portions may interact or respond to one another.

STRUCTURING RAW DATA

Armed with a plan, the next best place to start developing a tool is with the data itself. Whether you pull the data yourself out of the DATIM Data Genie or request the data through a SQL pull from someone in the Data Management Group, the data should be structured in a similar manner where you have your observations by row and indicators as your columns.

Figure 1: Raw data structure

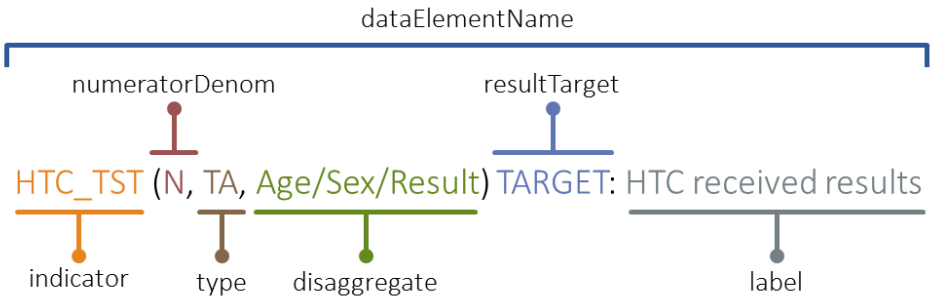
	A	G	H	I	L
1	orgLevel3Name	dataElementName	categoryOptionComboName	indicator	disaggregate
2	Angola	HTC_TST (N, TA, Age/Sex/Result): HTC received results	(Negative, Female, 15-19)	HTC_TST	Age/Sex/Result
3	Angola	CARE_CURR (N, TA, Age/Sex) TARGET: Received CD4, VL or Assessment	(Female, 15-19)	CARE_CURR	Age/Sex
4	Angola	TX_CURR (N, TA, Age/Sex) TARGET: Receiving ART	(Female, 5-14)	TX_CURR	Age/Sex
5	Angola	HTC_TST (N, TA, Age/Sex/Result) TARGET: HTC received results	(Negative, Female, 15-19)	HTC_TST	Age/Sex/Result
6	Angola	TX_CURR (N, TA, Age/Sex) TARGET: Receiving ART	(Female, 15-19)	TX_CURR	Age/Sex
7	Angola	CARE_CURR (N, TA, Age/Sex) TARGET: Received CD4, VL or Assessment	(1-4, Female)	CARE_CURR	Age/Sex
8	Angola	CARE_CURR (N, TA, Age/Sex) TARGET: Received CD4, VL or Assessment	(<1, Female)	CARE_CURR	Age/Sex
9	Angola	CARE_CURR (N, TA, Age/Sex) TARGET: Received CD4, VL or Assessment	(10-14, Female)	CARE_CURR	Age/Sex
10	Angola	CARE_CURR (N, TA, Age/Sex) TARGET: Received CD4, VL or Assessment	(5-9, Female)	CARE_CURR	Age/Sex

Data structure becomes extremely important when working with a tool or visualization that is dynamic or gets updated periodically. Even trying to start the process with fictitious data can be difficult without knowing how the data will be structured; you will likely spend a lot of time rebuilding the formulas within your tool to work with the actual data structure.

When developing a dashboard, there are a number of key indicators you will likely make use of when working with PEPFAR MER data.

- orgLevel*Name – Each level of the OU hierarchy is assigned a number starting with 3. The national/regional level starts at orgLevel3Name for all OUs. The organization levels vary country to country, but as you increase on the numeric scale you identify smaller and smaller sub national units (SNU). It's a good idea to know at what level each OU is setting target at, i.e. their priority level SNU. This information can be found on the DATIM Support page.
- uidlevel* - UID is the unique identifier used in spatial mapping, but also is good to use to ensure you are not aggregating two different SNUs in different regions , i.e. a country could have two separate districts called Centerville, one located in the North Province and one in the South Province.
- period – Knowing the time period you are drawing on is a key component of your dataset. It is important to realize that the output from DATIM is reported in calendar year unlike most of the rest of PEPFAR reporting. So, the second quarter of fiscal year 2016 (January to March 2016) would be reported as "2016Q1". Most of the major indicators are reporting on a quarterly basis, but some are reported only semi-annually or annually. Check out the MER Indicator Quick Reference Guide on DATIM Support for more details. This reference guide will also provide you with guidance on how to report indicators at the end of the year, e.g. aggregating all four quarters or taking the fourth quarter's results. Lastly, if you are working with targets, they are reported once a year. The period for 2016 targets would be recorded as "2015Oct"
- dataElementName – This field is a bundle that provides all data element pieces concatenated together: indicator, numeratorDenom, type, disaggregate, resultTarget, categoryOptionComboName. It is often much more useful to use each of the data elements pieces rather than this one that combines them all together.

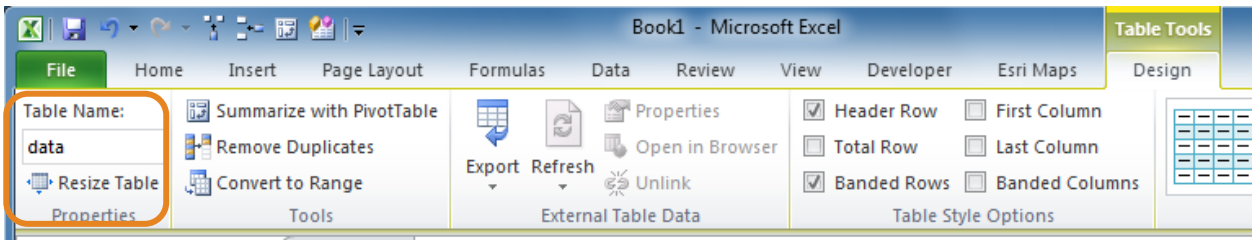
Figure 2: Pieces of the dataElementName



- indicator – This element provide which indicator is reported in the line, e.g. HTC_TST, PMTCT_ARV, TX_CURR.
- numeratorDenom – This element is either "N" for numerator or "D" for demonimator. You will typically be using the numerator for most of your work.
- type – This element is not frequently used as you tend to combine DSD and TA results.
- disaggregate – This element comes in a number of shapes and sizes, varying indicator to indicator. Indicators can be reported as the full numerator or can be broken out into various parts such as by age, sex, result and combinations of those. It is also important to note that these disaggregates can also have aggregates such has "<15" and "15+"
- resultTarget – This element indicates whether the data element is result or target. "TARGET" is written into the data element name if it's a target; the absence "TARGET" indicates the data element is a result.
- label – This element describes the indicator more clearly rather than just using the abbreviations. This element will be rarely used in typical analysis.
- categoryOptionComboName – This element is tied to the disaggregate. If there is no disaggregate, i.e. it's just the numerator entered, the categoryOptionComboName would be "default". If the disaggregate were Age/Sex, the categoryOptionComboName could look like "(Female, 5-14)".It is important to note that the order of the categoryOptionComboName can vary across and within indicators, so you can see "(Female, 5-14)" or "(1-4, Female)".
- value – This element provides the target or result value.

With the relevant indicators loaded into Excel, you should set up the data as a Table. To do so, you will need to highlight all the rows and columns containing data (Ctrl + *) and navigate on the top ribbon to Insert > Tables > Table Structuring.[1] After setting up the table, you should navigate in the ribbon to Design > Properties > Table Name, and change it to "data".

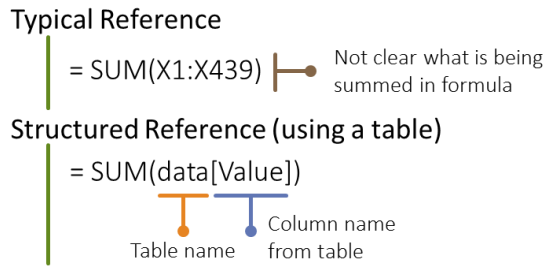
Figure 3: Changing table tame



[1] To remove the table formatting, make sure you have selected a cell within the table and navigate on the ribbon to Design > Table Styles > Clear.

The benefit of setting up your raw data as a table and renaming it comes from the ability to use structured references. This adjustment is incredibly important because it allows you to more easily reference your data in a formula and improves the readability of your formula for others.

Figure 4: Utilizing a structured reference



Now that you have your data structured using a table, you have the proper base for starting to build your tool.

EXCEL SETUP

The table that you have setup in Excel will form the basis for the rest of your file. You can call this tab "Raw Data." In addition to the raw data tab, there are a few other useful tabs to include.

The first additional tab to include is the "Info" tab. This tab will include key information about the data being displayed, including any caveats or notes your audience should know about the data. Equally important, you should always include the date and location of your data pull so your audience will know how recent the data is and more easily identify if there are known issues with the dataset. This tab is also a good place to place the ICPI logo if it is not displayed elsewhere and to provide any relevant contact information for the tool developers.

Another good tab to include in the file is a reference tab, which can be called "rs" for Rosetta Stone. The purpose of this tab is to provide (a) a crosswalk between items, e.g. regions and countries, (b) the location of all lists to be used in dropdowns, or (c) any other relevant reference material called upon by the tool or necessary for the developer.

The last necessary tab or tabs to include provide your views of the data. The next section discusses how to use formulas and other key features to aggregate the data and tell a story via visuals or tables.

CALLING AND AGGREGATING DATA

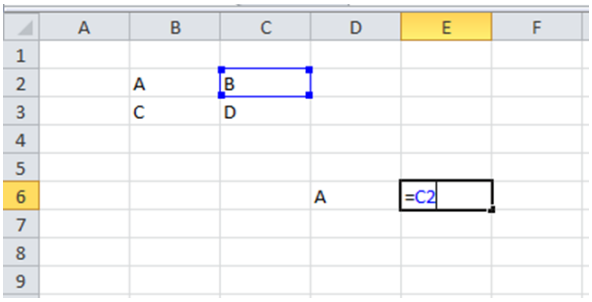
The bulk of the work comes when you have to start creating the tables and visuals for your tool. Working with Excel, there are often multiple ways to go about the process of "calling" or looking up data into a table to get the information you want to display. PivotTables are often one route and can be easy to setup, but can suffer from their bulky structure, their need to be refreshed, and random bugs. Although it can take a bit time more to setup and validate, another method would be to look up the data through formulas. These formulas can be a bit complex, but benefit from their ability to be dynamic and malleable. Setting up your raw data as table will make will decrease some of the complexity and improve readability of your formulas. Below are some key features and functions from Excel which will be the building blocks of your tool.

Features to be Familiar With

Before diving into the formulas, there are some key features to be aware of and use heavily in your tool.

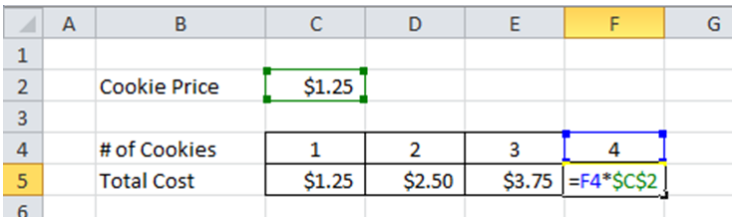
- Data Tables – setting up your data as a data table was referenced Structuring Raw Data section. As mention in that section, converting your data to a data table in excel will allow you to more easily reference your data in formulas and make your formulas more legible. If you are updating the tool on a frequent basis, you can just append new data to the bottom of the table, without the need to update any of your formula references elsewhere in the file. Additionally, setting up a table allows you to easily sort/filter, format, and carry formulas across all rows.
- References – cell references are critical for working with formulas in Excel. Below are four types of references.
    - Relative – relative references are the default in Excel. They are called relative because they move relative to the cell your formula is in. For example, in Figure 5, cell D6 references B2. If you copy this formula over one cell to the right, the reference will move one to the right, referencing C2.

Figure 5: Relative reference



    - Absolute – when using an absolute reference, this essentially "locks" your reference cell regardless of where you move the formula. In Figure 6, the formula in row 5 uses a relative reference (the cell above) for quantity and an absolute reference (C2) for the price. To get an absolute reference, you just need to hit F4 once after you select a cell. The benefits of using an absolute reference are that if the price of a cookie changes, you only need to update it in one place without having to adjust all the formulas to reflect the new price of cookies.

Figure 6: Absolute references



    - Mixed – Often times you want to have a mixed reference, which is a combination of the relative and absolute, allowing you to lock a column or row. To get a mixed reference, hit the F4 key two or three times to lock either the row or column, respectively. In Figure 7, to get the revenue (price x quantity), you can use one formula and copy it over to all the cells in the table. We need to lock in the column for the price (C) and lock the row (3) which contains the quantity. The formula in D4 would be =D$3*$C4. As you

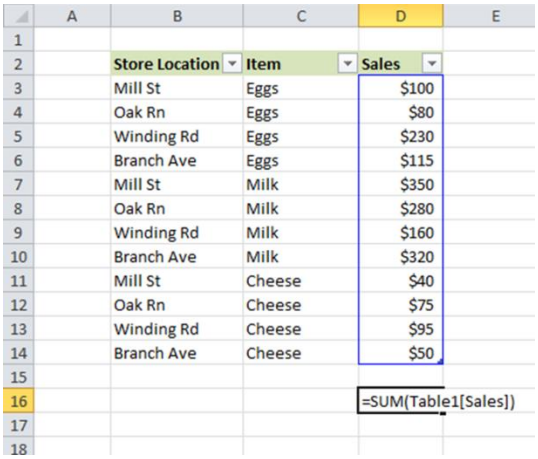
copy the formula over to the other cells in the table, the formula holds the quantity fixed going down a row while adjusting the price and holds the price constant while moving horizontally as it adjusts the quantity.

Figure 7: Mixed reference

	A	B	C	D	E	F	G
1							
2				Quantity			
3			Price	1	2	3	4
4		Cookie	$1.25	$1.25	$2.50	$3.75	$5.00
5		Candy Bar	$1.00	$1.00	$2.00	$3.00	$4.00
6		Ice Cream	$3.25	$3.25	$6.50	$9.75	=G$3*$C6
7							

- Structured – Structured references can only be used when working with tables. They act like mixed references but have the added benefit of using clear labeling in the formulas. For instance, to get the total sales from the table in Figure 8, instead of writing "=SUM(D3:D14)" you can use the structured reference, identifying the table and column heading (in brackets), "=SUM(Table1[Sales])".

Figure 8: Structured reference



	A	B	C	D	E
1					
2		Store Location	Item	Sales	
3		Mill St	Eggs	$100	
4		Oak Rn	Eggs	$80	
5		Winding Rd	Eggs	$230	
6		Branch Ave	Eggs	$115	
7		Mill St	Milk	$350	
8		Oak Rn	Milk	$280	
9		Winding Rd	Milk	$160	
10		Branch Ave	Milk	$320	
11		Mill St	Cheese	$40	
12		Oak Rn	Cheese	$75	
13		Winding Rd	Cheese	$95	
14		Branch Ave	Cheese	$50	
15					
16				=SUM(Table1[Sales])	
17					
18					

- Named Ranges – A very useful feature in Excel allows you to name ranges of cells. Like with the structured reference, using named ranges makes it more clear what information is being used within a formula. For instance, you have one cell named "ctry_select" which is the Operating Unit of focus in the table or visual. All formulas can use the "ctry_select" name and if the OU changes, none of the formulas need to be updated. To create a named range, you start by selecting the cell or group of cells you want to reference. You can add the name by navigating on the ribbon to Formulas > Defined Names > Define Name.
- Text Validation/Dropdown Menus – When designing a tool for use across multiple, it is often useful to have one set of visuals/tables that is dynamic and changes as you adjust the OU of focus. One way to do this is to develop a dropdown selection of OUs for the viewer to choose from. This list of OUs should be stored in the "rs" tab. To create a dropdown, click on the cell where you want to have the dropdown and then navigate on the ribbon to Data > Data Validation. Once there, you need to change the "Allow" box to "List" and add in the named range for your Source.

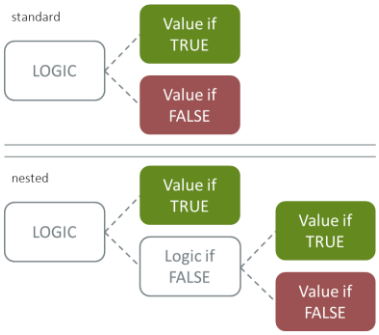
- Remove Duplicates – When creating named range, like a list of OUs, you often start with all the observations in your dataset and whittle it down to unique observations. One way to do this in Excel is to the use the Remove Duplicated feature. You can pull the full list of countries into your "rs" tab and then, with the list selected, navigate on the ribbon to Data > Data Tools > Remove Duplicates. You will be left with a unique list.

Logic Formulas

Using the following formulas will assist you create more complex formulas by allowing you to subset your dataset or adjust to errors.

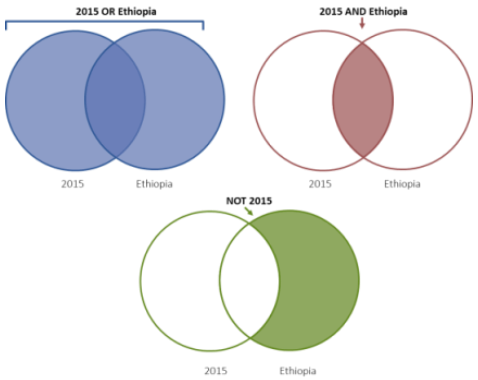
- AND – determines whether multiple inputs are true [AND(logical1, [logical2], ...)]
- OR – determines whether at least one of the inputs is true [OR(logical1, [logical2], ...)]
- NOT – reverses the logic, e.g. NOT(1+1=2) = TRUE [NOT(logical)]
- CONCATENATE/& - combines multiple items together into one cell as text; this can be done via the CONATENATE function or using an "&". All text, i.e. non-cell references, need to be surrounded by quotes, e.g. = "Date: " & TODAY() → Cell would read: "Date: April 7, 2016" [CONCATENATE(text1, [text2],…)]
- IF – These logical formulas become more useful when combined with IF statements. Essentially, you are giving Excel orders based on the output of logic. For example, if the value of a cell is greater than 100%, write ">100%", otherwise use the value. You can also "nest" IF statements within other IF statements (up to 64 times!). It is often useful to combine the other logic operators (AND, OR, NOT) in the IF statements. [IF(logical_test, value_if_true, [value_if_false])]

Figure 9: IF Logic, Standard and Nesting



- IFERROR – tells Excel what the cell should display or value it have if the function produces an error; this is useful where you may not want Excel to display "#N/A" when it experiences an error in the calculation [IFERROR(value, value_if_error)]

Figure 10: Visualizing logical expressions – OR, AND, and NOT



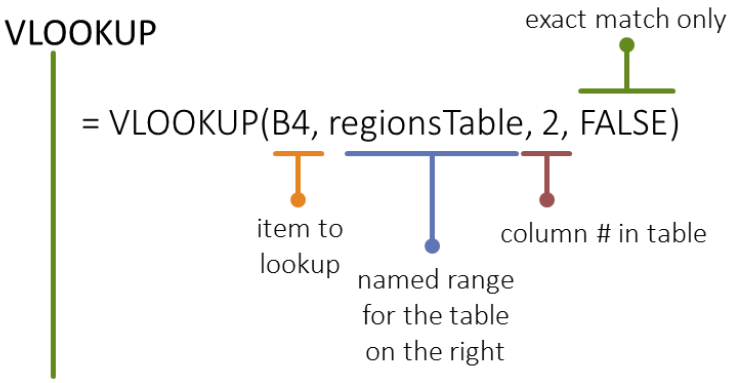
Lookup Formulas

Armed with a better understanding of some of Excels features and logic operators, it is now time to dive into the meat of creating dynamic tools: lookup functions. Below are a number of functions that prove to be quite useful for pulling data from the raw data to aggregate or reference in the table.

- SUMIFS – This is an extremely useful function when working with the data table in Excel. The data you have exported likely has multiple observations for an indicator within the same SNU; SUMIFS allows you to aggregate while placing constraints on the data, e.g. you want to know how many individuals under 15 years old in the West Providence of Country X are on treatment in Q4 of 2015. See the appendix for more on working with SUMIFS equations. [SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], ...)]
- VLOOKUP – The VLOOKUP, or vertical lookup, function can be used when you are looking up a value from one table that is an exact match to an item in another table. For instance, you could have a table that countains all our OUs with other information. You can then use the VLOOKUP function to find the region in another table that contains the OU names and regions. When using VLOOKUP, the table you are mapping onto should only contain one, unique observation of the value you are looking up and it needs to be an exact match. In the example below, you have a list of the top ten largest countries in terms of population. What you would like to include is the region these countries are located. Rather than manually looking up the region for each country from the list on the right, you can use the VLOOKUP function to populate the table with regions. Note that the value being looked up in the table must be the first item in the lookup table. So, in the example below, you are looking up the region based on the country name. Your table array i.e. the area where you are looking up another value, must be in the first column. [VLOOKUP (lookup_value, table_array, col_index_num, [range_lookup])]

Figure 11A & B: VLOOKUP to find a region and VLOOKUP structure

	B	C	D	H	I
1	POPULATION TABLE			REGION LOOKUP TABLE	
2	Country Name	Region	Population (2014)	Country Name	Region
3	China	East Asia & Pacific	1,364,270,000	Aruba	Latin America & Caribbean
4	India	=VLOOKUP(B4,regionsTable,2,FALSE)		Afghanistan	South Asia
5	Indonesia		254,454,778	Angola	Sub-Saharan Africa
6	Brazil		206,077,898	Albania	Europe & Central Asia
7	Pakistan		185,044,286	Andorra	Europe & Central Asia
8	Nigeria		177,475,986	United Arab Emirates	Middle East & North Africa
9	Bangladesh		159,077,513	Argentina	Latin America & Caribbean
10	Russian Federation		143,819,569	Armenia	Europe & Central Asia
11	Japan		127,131,800	American Samoa	East Asia & Pacific
12	Mexico		125,385,833	Antigua and Barbuda	Latin America & Caribbean
13	Source: World Bank, WDI			Australia	East Asia & Pacific



- INDEX-MATCH – INDEX-MATCH takes VLOOKUP to the next level (its actually two different formulas). It is similar to VLOOKUP in that you are looking to match a value in an array and return the result. One of the major benefits is that it makes the row dynamic as it looks to match the name rather than you entering a static row number that could change. Another benefit is that you can look up values to the left of your matched column, not just to the right. And a last point to note, INDEX-MATCH takes up a lot less processing power than VLOOKUP, so your tools will respond more quickly and are less likely to freeze up on your user.

Figure 12: INDEX & MATCH Equation Structures

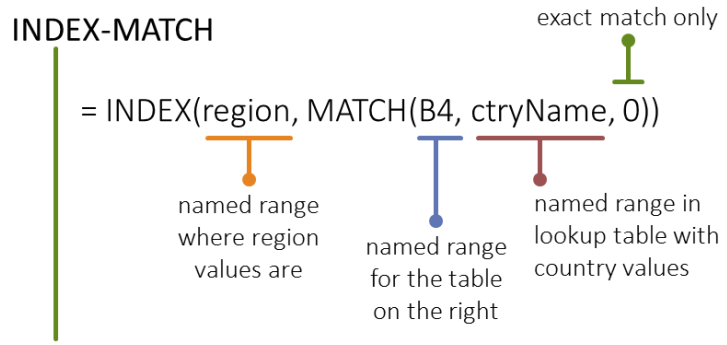
INDEX & MATCH Structures

```
=INDEX(array, row_num, [column_num])
=MATCH(lookup_value, lookup_array, [match_type])
```

- If we were to use INDEX-MATCH instead of VLOOKUP in Figure 12A, we would use the equation found in Figure 13B.

Figure 13A & B: INDEX-MATCH

	A	B	C	D	E	F	G	H	I
1		POPULATION TABLE						REGION LOOKUP TABLE	
2		Country Name	Region	Population (2014)				Country Name	Region
3		China	East Asia & Pacific	1,364,270,000				Aruba	Latin America & Caribbean
4		India	=INDEX(region,MATCH(B4,ctryName,0))					Afghanistan	South Asia
5		Indonesia		254,454,778				Angola	Sub-Saharan Africa
6		Brazil		206,077,898				Albania	Europe & Central Asia
7		Pakistan		185,044,286				Andorra	Europe & Central Asia
8		Nigeria		177,475,986				United Arab Emirates	Middle East & North Africa
9		Bangladesh		159,077,513				Argentina	Latin America & Caribbean
10		Russian Federation		143,819,569				Armenia	Europe & Central Asia
11		Japan		127,131,800				American Samoa	East Asia & Pacific
12		Mexico		125,385,833				Antigua and Barbuda	Latin America & Caribbean
13		Source: World Bank, WDI						Australia	East Asia & Pacific
14								Austria	Europe & Central Asia



ADDITIONAL ELEMENTS FOR SUCCESS

Part of creating an effective dashboard or tool in part comes from your use of formulas, but it also hinges on a number of other design elements. When building your tool, you should keep the following items in mind.

Consistent color scheme

At this time, ICPI does not have a color scheme to use when developing your tool or visualization. This fact can be both extremely useful since you are not bound to any set of colors, but it also has the downside that your colors can be all over the place. Although the colors may different from one tool to the next, it is essential that you stick to a consistent color scheme within your product. A good range of colors could be in the range of three to six. Whatever you do, you should avoid using the default colors in Excel. Graph using the default colors come across as lazy. At a minimum, you can explore the other schemes found in Page Layout > Themes > Colors. The internet is full of great examples of color palette you can use and recreate in your own tool. Some good places to start are Color Brewer or Adobe Color.

Figure 14: MS Office default color scheme

Fonts

Much like color, you should try to use a font scheme that varies from the default or typical font families, like Times New Roman or Arial. Take some time to play around with different fonts and find some that work well with the mood/feel of the tool and the size you are using. Try to limit the number of fonts to no more than three and make sure they all work together. Think about use of bold, italics, and size (in addition to color) for drawing attention to key pieces of information.

Viewing size

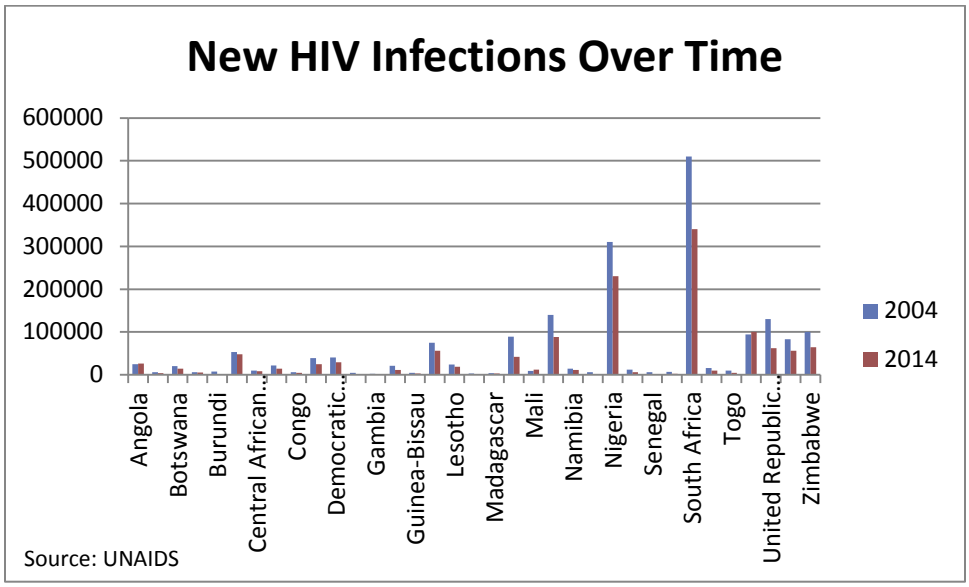
At the beginning when you were sketching out your purpose and overall structure of your tool, you should have brainstormed how it will be used. The key questions to consider is how will the end user interact with your tool? This decision will influence how you structure your dashboard/visual. If it just a graph/ table or intended to be printed out, make sure it fits to one page. Alternatively, if your goal is to construct a tool for the user to interact with on their computer, think about designing the layout around fitting on a standard screen size. Minimize the amount of scrolling the user will have to do and place key elements, visuals and/or text is prominent places on the screen, i.e. the upper left hand corner (as we read left to right starting at the top). Vary the size of your visuals and text to create contrast and draw a viewer's attention to important information.

Minimize excess

When working with a lot of data, it can be tempting to develop numerous tables and charts to highlight multiple items. Being a minimalist is a crucial part of effectively communicating your point. Rather than creating multiple figures and tables, think about the best way to show your main point(s) without overwhelming the viewer. Following the points avoid will help with this task, but really being judicious with your space (and white space), color, font, size, and supplemental text will go a long way.

Having too many items on your visualization can be distracting and obscure your overall message. This point also holds true within each of your visualizations. Take a look at Figure 15A for example. The figure looks at New HIV infections in Sub-Saharan African countries at two points in time. The clustered column chart simple graphs the data. What are improvements could be made to the Excel default figure created?

Figure 15A: New HIV infections (default view)



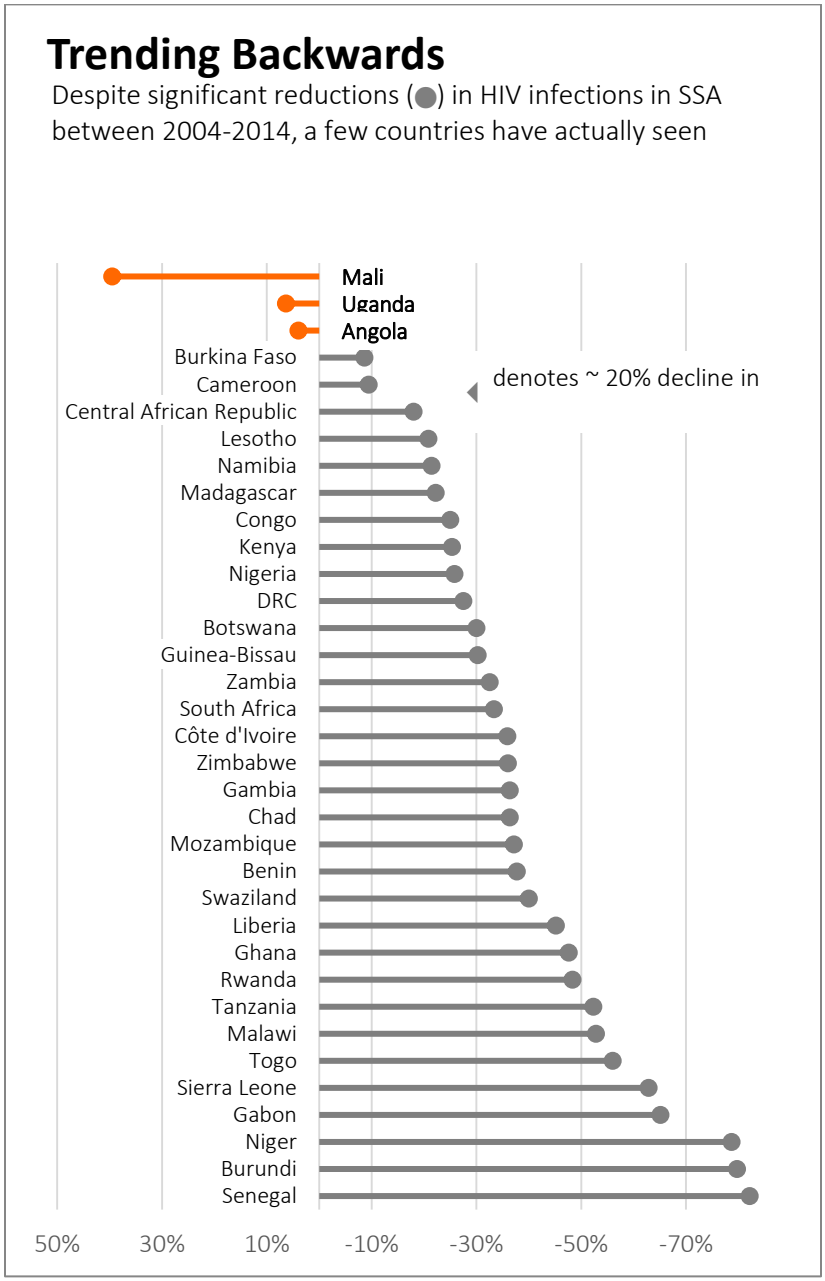
Issues/Improvements to Figure 15A

- When looking at this figure, it is far from clear what the **intended message** the viewer should come away with. You can see changes between the two periods, but is this figure trying to convey overall success of programming? Crafting a strong title and even subtitle can help convey the message you want to present when viewing the data.
- Another problem here comes from the **size**. Much of the data is obscured due to the size of South Africa's large HIV infection rates. One option would be to look at the change (growth/decline) over this period rather than the raw numbers. Additionally, the width of the individual bars could be increased so they are not so thin, decreasing some of the white space in the figure.
- When possible, you should try to **integrate the legend** into the figure rather than including it off to the side or on the bottom. Although there are only two bars, the viewer still has to do work to go back and forth between the graph and the legend to figure out which time period they are looking at. The more bars/colors, the more work you impart on the viewer.
- A potential improvement in the graph comes from labeling. Listed countries along the x-axis as is done in the figure makes it difficult to read. Where possible, you should **keep any text horizontal** making it much easier to read. A further point is that this setup leaves out half of the countries visualized in the figure. Solutions include switching the graph to a horizontal bar graph or limiting the scope of countries visualized.
- On a related topic to labeling is the ordering of elements. Often times in charts and figures, you will find the elements ordered alphabetically. Although this may make it easier to find a particular element, like a district, you should think about a **logical ordering of elements**. In the graph, rather than listing the countries alphabetically, it could have been sorted by HIV infections in 2004 or 2014 or even by the growth rate.
- Another "lazy feature" in the graph can be seen in the **numbering** on the y-axis. At a minimum, the numbers should include a comma to make them easier to read. Since all of these numbers are in the hundreds of thousands, they can be shorted to just a single digit with a label noting the values are in 100,000s.

- In addition to these issues, the graph also suffers from **"chart junk,"** i.e. excess/unnecessary feature in the visualization that detracts or draw attention away from key feature or the overall message. Examples of chart junk include bright colors, bold feature, tick marks, grid lines, 3D, and shading. For instance, in this graph, you could remove tick marks from around the axes, the y-axis line could be removed, and the remaining lines (x-axis and gridlines), could be kept, but changed to a lighter color so as not to draw attention.

Below is another graph (Figure 15B) that takes the same data as the figure above, updating it according to the bulleted notes.

Figure 15B: Updated graph of new HIV infections



Sharing

As mentioned earlier, one of your first steps is to determine the purpose of the tool and vision for the end product. If the goal is to have the Excel file shared with the user so they can interact with the tables and charts (as opposed to a static visual that could go in a report or presentation), the developer should consider a few items.

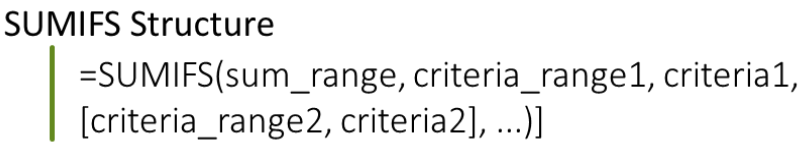
- Minimize clutter – Limiting clutter keeps popping up because it's an essential part of creating effective and visually pleasing designs. When sharing with others, make sure to limit the number of tabs, keeping the tab(s) for the user to browse and explore easily accessible and to a minimum. If the file requires having numerous (+5 tabs), include a navigation or table of contents tab as the "home screen." You can create links to (cells in) other tabs by right clicking on a cell or shape and selecting Hyperlink (or hitting Ctrl+K). Any tabs that used for background processing, dropdown lists, raw data, etc. should be hidden so as not to burden the viewer and keep their attention to the visuals you have created. To hide a tab, right click on the tab and select "Hide." Additionally, you should hide all unused ranges so the navigation is limited to just where the displays are. This can be accomplished by highlighting all of the columns/rows from the header, right clicking the header and selecting "Hide". Lastly, as mentioned earlier, you should think about containing the view to size of standard computer monitor so that anyone viewing (~13"-14").
- Protecting tabs – In Excel, you have the option to "Protect" tabs or the overall workbook. The benefit here is that you can create a polished look for your tool/dashboard by protecting it and limiting the user from clicking/viewing/editing the contents of cells or seeing underlying formulas. If you need the user to interact with portions of a tab, you can "unlock" when the sheet is protected. To unlock a cell when the sheet is protected, right click on the cell > Format Cells > Protection > uncheck the box labeled "Locked." When you protect a sheet, you have the option of adding a password. A best practice (unless the data is sensitive) is to protect the sheets and workbook without putting on a password.
- File Size – When working with large datasets or developing highly intricate/dynamic visualization, be wary of your file size. As your file grows in size, it will likely be bogged down in processing or may be difficult to share. Emailing files over 20MB may be blocked by some agencies.
- Security – Access data from sites such as PEPFAR.net and DATIM.org are limited to users within the USG who have a need to access the data. Some of the data, especially pertaining to key populations is sensitive and should be treated carefully. As soon as the data is exported from these sites, the data cannot be protected in the same way and the onus is on you to ensure you are sharing the information with people who should have access.
- View/Grid lines – Another feature in Excel to make use of when sharing a workbook as a tool/dashboard is to remove the gridlines. The act as unnecessary "chart junk" within your workbook. Removing the grid lines makes the tool look cleaner and formal.
- Info tab – And as a good measure, any time you are sharing your dashboard/tool, you should include an info tab that contains the information with any notes about the dataset, when and where the data was pulled, any caveats, etc.

APPENDIX

Using SUMIFS of Different Complexities

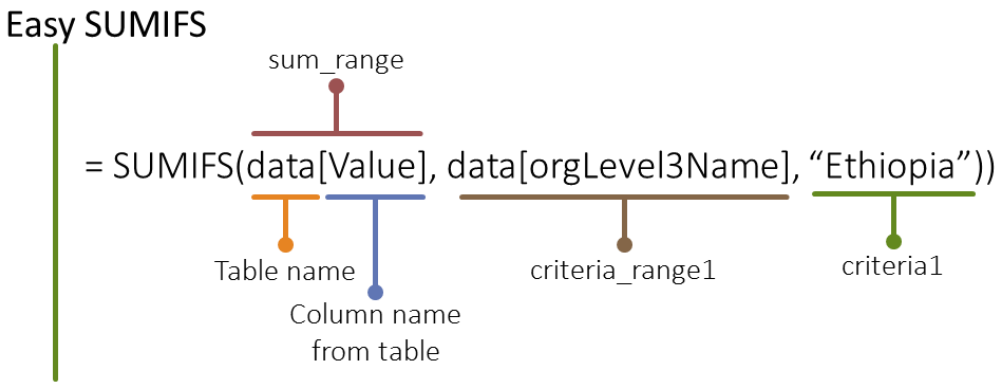
The SUMIFS function is very versatile which provides quite a bit of range when trying to look up and aggregate values in a table.

Figure A1: SUMIF Structure



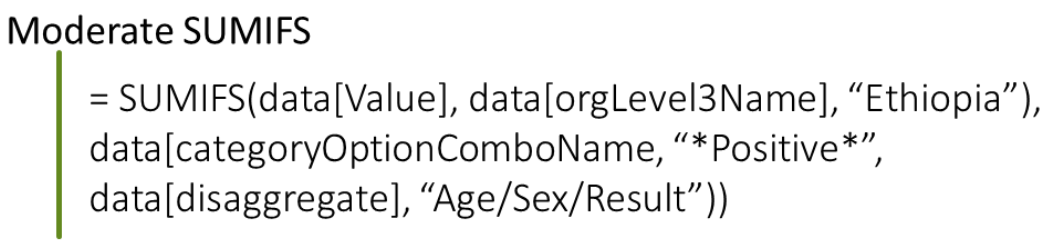
EASY – To start with, you can use SUMIFS to aggregate pretty easily. Essentially, you identify the range you want to sum and then can on as many criteria ranges and criteria pairs (up to 127, so quite a few). For instance, let's say you want to know how many people were tested for HIV (HTC_TST) in Ethiopia. You would use the formula below in Figure A.1 to identify the range you want to sum, the criteria range, and criteria.

Figure A2: SUMIFS, Easy



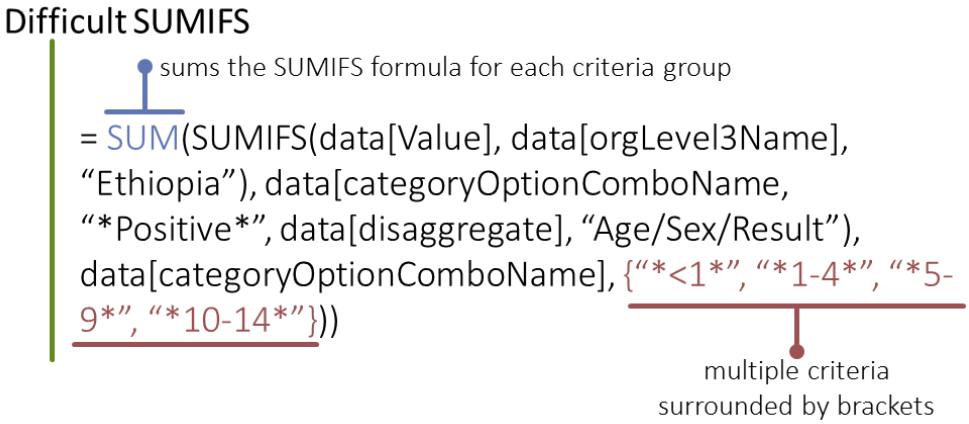
Moderate – Our SUMIF statement get a bit more complex when we add in a piece like wanting to know about positive tests. It's a similar structure as in the easy example, we just need to tack on another range/criteria pair to the end. The difficulty of this step comes from having to aggregate positives which are in the categoryOptionComboName indicator in the DATIM dataset, but comes "packaged" with other information on age group and gender e.g. "(Female, 15-19, Positive)". Rather than having to know all the text combinations that include Positive to get an exact match, you can you a asterisks (*), which is a wild card used to search for partial match. To find all the positives, you would use "*Positive*". One additional complication is that just adding this range/criteria is would end up doing some double counting, as it would include both finer (<1, 1-4, 5-9,…) and grosser age disaggrations (<15, 15+). In the formula, we would need to specify that we wanted to look at one or the other (preferable finer age disaggregations).

Figure A3: SUMIFS, Moderate



DIFFICULT – Taking the SUMIFS to the next level, what if we wanted to know about testing for pediatric patients. Since finer age groups are typically used, you would need to sum up the values in the following age bands: <1, 1-4, 5-9, and 10-14. An inefficient way to perform this task would be to write up 4 different SUMIFS formulas and add them together. Instead, you can use SUMIFS as a type of array function. Essentially what you are doing is adding multiple criteria like an OR statement. In the criteria portion, you would need to separate out the different age groups by comas and contain them all within curly brackets. If you created this formula and hit enter, Excel would only return the sum of the first age group, <1. To perform this function on each group and aggregate the results, you need to add a SUM function to the beginning of the equation.

Figure A4: SUMIFS, Difficult

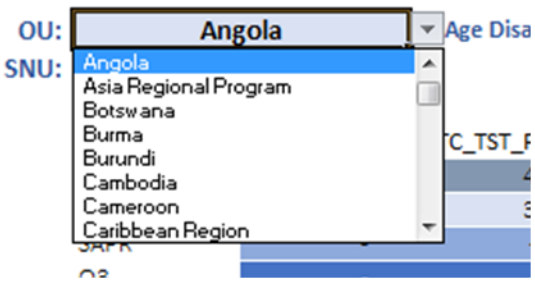


Dropdown Menus

Creating a dropdown menu of options allows for an interactive user experience in Excel. The end user can adjust the views by operating unit, indicator, timer period, or age band to name a few examples. Creating a drop down menu is relatively straight forward. To create a dropdown, click on the cell where you want to have the dropdown and then navigate on the ribbon to Data > Data Validation. Once there, you need to change the “Allow” box to “List” and add in the named range for your Source.

Figure A.5: Dropdown menu and source (using a named range)

front end dropdown selection



list/named range on backend

	OU
1	ALL ACT
2	Angola
3	Asia Regional Program
4	Botswana
5	Burma
6	Burundi
7	Cambodia
8	Cameroon
9	Caribbean Region
10	Central America Region
11	Central Asia Region
12	Cote d'Ivoire
13	Democratic Republic of the Congo
14	Dominican Republic
15	Ethiopia
16	Ghana
17	Guyana

Dynamic Named Ranges

TBD