Consistency VBA: Users Manual

The Code:

The purpose of this VBA program is to read a text file filled with output values from a script. It then calculates the averages and standard deviations of all test variables in the workbook. Then the newest test data is tested and if enough values are outside of allowed limits a file would be created that shows the worksheets and cell that the events occurred in.

Getting Started:

For those unfamiliar with VBA (Visual Basic for Applications) it is a type of programming application that allows the user to manipulate excel and other files. To access (at least with excel 2007) you have to open an excel file and check to make sure the developer tab is in the top right. If you cannot see the developer tab click on the windows tab at the top left, and at the bottom of that menu click the “excel options” button. Then click the “Show developer tab” box then click “OK”. The developer tab should now be visible in the top right. Clicking on the developer tab shows Visual Basic, and clicking on the Visual Basic button opens the application.

Using the Consistency Code:

Note: ALWAYS SAVE BEFORE RUNNING THE PROGRAM IN CASE THERE IS A PROBLEM TO AVOID LOSING DATA.

This code will also not work if you try to use more columns than excel has available. Excel goes into column XFD, which is over 15000 columns in. That’s a lot of columns so you should be fine. However if you use more than 15000 you might have to split the file in half to run it.

Once the application is open you want to click on insert and add both a user form and a module. To the left you’ll see a tree of files. Click on the user form listed under your workbook and hit enter to select the form. Then use the toolbox under view to add a command button. Then click view 🡪 code to see the code under the user form and copy the user form code into the box under the command button sub (replacing everything that is already in the code). After copying the code, open the module under your workbook and copy the module code into the box (replacing everything that is there).

If you want to you can take an old excel file with the code in it, delete the worksheets so that it starts out as a blank workbook then save it under a new name. Now you have two workbooks both with the working consistency code.

Now you should be free to run the code by reopening VBA and selecting the user form. Then click run 🡪 run sub/user form. This would pop up the user form. Click the command button and the program should run.

If an error comes up saying macros are not enabled this can be fixed under the developer tab. Click macro security button, enable macros, save the file and reopen it.

The User Form:

The user form has six (6) command buttons and nine (9) text boxes. Each of these is labeled.

Command Buttons

1. *Script Run - This* button will run a text file created from running a script from the scripted controller. It is designed so that it will properly write into the excel file with the roster printed in a vertical manner. Also the file needs to be indexed. The index is set up so that every time the script re runs the procedure the first number in the column is a zero (0), and every other time step starts with a one (1) in the first column. Every time a zero appears new worksheet is created in the file and the averages and standard deviations are recalculated. The script file that this code runs is in the form of:  
   Roster

Variable name 1

Variable name 2

…

Tab delimited data.

If you try to run the file and it is headed with a variable that consistently has the value zero then you will have errors printing the worksheets.

1. *Solution File Run - This* command button will run a text file that was saved as a solution in HumMod.exe. These are usually very large and take longer to run than the script file. Solution run is also not coded with the index function.
2. *Click to Graph* - This button will graph a scatter plot, on as many worksheets as you would like, for each variable you list (setting domain, ranges, and worksheets will be discussed later). The graph will often try to graph the variable name and those often appear as vertical point on the chart. The lines that are graphed however are accurate despite the extra points that were graphed.
3. *Compare All -* This button will take the most recent sheet, recalculate averages and standard deviations, and then will run the comparison command. This will take every cell in every data worksheet and will compare them to the averages and standard deviations. If 16% of the cells are outside of average plus/minus double the standard deviation, or 5% of the cells are outside of average plus/minus triple the standard deviation, then it will record the worksheets and cells and print them into a file that will be placed in the same folder as the excel workbook. The script and solution run buttons will compare, but instead of comparing all of the pages it will only compare the last ten worksheets. Because of the index function, several sheets won’t be marked because if you index thirty (30) runs only ten of them are marked in the event file. The compare all button will allow you to mark any cells that might be new, but not in the last ten sheets. There is a macro called comparison that contains code to compare all the sheets (commented out because that function is available under the command button), the last ten, which is active code, and the last 100 sheets which is coded out and not available in another function. If you wish to compare the last 100 sheets all you have to do is delete the apostrophes (‘) so that it won’t read the code as a comment. While un-commenting the code if an error appears that means that was originally a comment and must have the apostrophe.
4. *Clear Workbook* - This button allows you to start over with a blank workbook if you need to restart for any reason. It will delete all the graph, data, and worksheets leaving you with a blank workbook with only one worksheet titled “Sheet1”.
5. *View Worksheet -* This button allows you to view a worksheet that you type into a text box without having to stop the VBA program from running.

Text Boxes

Text Boxes will reset every time you restart the program. If you plan to always use the same pathways/filenames/etc. then you can write them in the text boxes before running the program. Those will become the default text box values.

1. *Script data pathway* - In this textbox list the pathway to the text file created by a script. And example pathway is:

C:\Users\Desktop\

Make sure that the pathway ends in a backslash (\) otherwise the code cannot find the proper file.

1. *Data File Name* - This is the name of the file that you want to open whether it was created by a script or saved as a solution through HumMod.exe. Make sure the name is exact, an example is:

Test.txt

So far this code has only been tested with text files. If you want to try reading other files (such as a word file) I cannot promise it will work.

1. *Event File Name* – This text box allows you to name the file that events (from the comparison function) are recorded in. Whatever you write in this text box and the date will be the name of the file. For instance if you write:

Test

In the text box on June 28, 2012 then the file name will be

Test 06\_28\_2012.

1. *Worksheet Name (for Graphing)* – This text box contains the list of worksheets you want to graph on. All worksheets are named “Sheet” followed by a number. All you have to type in is the sheet number followed by a semicolon (;). So if you wanted to graph Sheet7, Sheet8, and Sheet9, you would type this in the text box:

7;8;9;

The last semicolon is very important. Without this semicolon either the last listed sheet won’t be read or the program will be stuck in an infinite loop and you have to manually close it.

1. *Workbook Name* – This text box is required to know which workbook you are working in. It is not used to choose the name to save the workbook. That must be done manually under the save as function in excel. Although this name is used to specify which workbook you are working in, occasionally the code will run in another workbook if you have multiple open. To avoid this make sure the workbook you want to work in is always active on your screen before opening/running VBA. Another solution is to have the workbook you are working in as the only open workbook on your computer. The workbook must be saved as macro enabled to make sure it is properly named with .xlsm. For instance if you are working in Workbook “Consistency” then what you write in the text box is:

Consistency.xlsm

1. *Solution File Pathway* – This text box performs the same function as script data pathway. This is the pathway the computer must follow to find the solution file. An example pathway is:

C:\Users\Desktop\

Again, the last backslash (\) is very important otherwise the program cannot find the file.

1. *Range (Exact Variable Names)* – This text box holds the variables you want to graph as the range. The variable names must be exact otherwise they won’t be graphed. This includes any spaces in the cell (For instance variable names from a solution run have a space before and after the word. If you do not include these spaces it won’t graph.) You may choose multiple ranges, and a new chart will be created for each variable you declare as a range. Every variable needs to be ended with a colon (:) otherwise it won’t be read by the code. An example range is:

CardiacOutput.CO:System.X:

The last colon (:) is important.

1. *Domain Variable* – This is the variable that you wish to be the domain. The domain will be this variable for every range you specified. The name must be exact, just like with the range. Example Domain:

System.X

1. *Worksheet (to view)* – This is the worksheet you want to view without having to close the user form. Just like with the graphing worksheet list all you have to write is the sheet number, but you can only have one sheet written in the text box at a time. For instance if you want to view Sheet7, you write this in the text box :

7