

Welcome to the SolverTable add-in

This add-in is a natural extension to the Solver add-in developed by Frontline Systems. SolverTable performs sensitivity analysis for an Excel optimization model, and, in most cases, its output is more relevant and understandable than the optional sensitivity output provided by Solver itself.

SolverTable is very easy to use, as I describe below. All it requires is an existing optimization model. That is, there must be a spreadsheet model already built, and Solver must be installed. This document explains how to use SolverTable, and it illustrates the procedure with several screen shots.

Sections of this Help file

- [Versions of SolverTable](#)
- [Loading SolverTable](#)
- [Unloading SolverTable](#)
- [An example file](#)
- [Running SolverTable](#)
 - [Creating a one-way table](#)
 - [Creating a two-way table](#)
 - [Being creative](#)
- [Other Notes](#)

Versions of SolverTable

I developed SolverTable in the late 1990s, and I have been enhancing and modifying it ever since. It has always presented a challenge because it invokes Solver, and Solver has some behind-the-scene idiosyncrasies that make it difficult to program for. This caused problems (for a few users, certainly not for all users) with SolverTable for Excel 2003 and earlier, and for Excel 2007. These problems are documented on my www.kelley.iu.com/albrightbooks/free_downloads Web site. To combat these, I rewrote the add-in for Premium Solver and Risk Solver Platform, two advanced versions of Solver, and renamed it SolverTablePremium. This version has worked fine for just about everyone, the reason being the different API (application programming interface) used by these versions of Solver. The only problem is that you have to purchase these advanced versions of Solver; they do *not* ship with Office. The current version of SolverTable was rewritten explicitly for Excel 2010, but it should work fine with Excel 2007. There are two problems you might encounter and should be aware of:

- 1) All versions of Solver through Excel 2007 (but not Excel 2010) were “smart” add-ins. This means that Solver doesn’t actually load into memory until it is invoked, that is, until you open Solver dialog box. (Frontline Systems did this to save on memory, but memory is hardly any issue on today’s PCs.) This caused a lot of problems with SolverTable. In my online help, I referred to “waking up” Solver to ensure that SolverTable would work

properly. You can fix this problem once and for all on your PC by editing the registry. Specifically, click on the Windows Start button, click on Run, type in **regedit**, and click on OK to get into the registry. Then navigate to HKEY_CURRENT_USER/Software/Microsoft/Office/12.0/Excel/Options. On the right, you will see an OPEN item that starts F/ “C:\Program Files” and ends with “Solver.xlam”. That’s the one you need to change. To edit it, just double-click on the OPEN item, and in the editor, delete the first 3 characters (/F). Now close out of the Registry Editor and you should be all set. That is, from then on, if Solver is checked in the list of add-ins (there is a Solver button on the Data ribbon), it will automatically load into memory when Excel opens, and SolverTable will work correctly. Again, this is *not* a problem with Excel 2010; the “/F ” is *not* in the registry item for Solver in Excel 2010. By the way, you can make the same change in Excel 2003 or earlier; the only difference is that the add-in file is Solver.xla, not Solver.xlam.

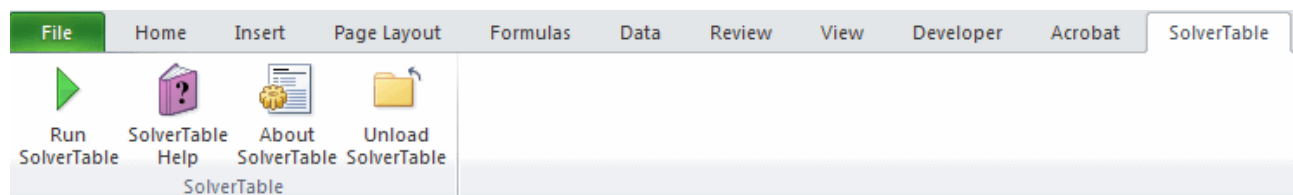
- 2) SolverTable needs to “find” the Solver.xlam file for its code to work. The only time this will fail in Excel 2007 or 2010 is a very unlikely situation: Solver is not loaded (there is no Solver item on the Data ribbon) *and* the Solver.xlam file is not in its default location. The latter would occur only if you purposely chose a different location for Solver when you installed Office. You can check this. The default location for Solver is C:\Program Files\Microsoft Office\Officexx\Library\Solver. But even if Solver.xlam is not in its default location, SolverTable will still work properly, provided that Solver is loaded.

Loading SolverTable

To load SolverTable, proceed as follows:

- 1) If you haven’t already done so, unzip the **SolverTable 2007_2010.zip** file to a folder on your hard drive. I recommend creating a SolverTable folder for this purpose under Microsoft’s Library folder, i.e., c:\Program Files\Microsoft Office\Officexx\Library\SolverTable, because this folder is in Microsoft’s “trusted” list. If you want to store the files somewhere else, e.g., D:\SolverTable, you should add this folder to the trusted list. To do so, Click on the File tab, then Options (in Excel 2010), or the Office button, then Excel Options (in Excel 2007), then Trust Center, then Trust Center Settings, then Trusted Locations, and then Add new location.
- 2) In Excel, load SolverTable into memory. There are two ways to do this. I suggest the first, but regardless of the method you use, you will know that SolverTable is loaded when you see a SolverTable tab with its own ribbon, as shown in Figure 1. (The SolverTable tab is placed to the right of all existing tabs.)

Figure 1



- a) Click on the File tab, then Options (in Excel 2010), or the Office button, then Excel Options (in Excel 2007), then Add-Ins, and then the Go button at the bottom. Here you see the familiar Add-ins dialog box from earlier versions of Excel. (**Note:** At least in the beta of Excel 2010, there is a much quicker way to get to the add-ins list—finally! Simply click on the Add-Ins button on the Developer tab. If the Developer tab isn't visible, you can make it visible, permanently, by clicking on the File tab, then Options, then Customize Ribbon, and checking Developer on the right.) If this is your first time to load SolverTable, you will need to click on the Browse button to find the **SolverTable.xlam** file. Otherwise, there should be a SolverTable item in the list of available add-ins. Just check its box, and click on OK. The advantage of this method is that SolverTable will open automatically every time you open Excel. In fact, it will keep opening until you uncheck SolverTable in the Add-Ins list.
- b) Alternatively, open the **SolverTable.xlam** file in the same way as you open any other Excel file. If this file is buried way down in some folder, this might not be very convenient, but you can accomplish the same thing by putting a shortcut to this file on your desktop so that it is just one click away. The advantage (or disadvantage?) of this method is that if you then close Excel and reopen it, SolverTable won't be loaded; you will have to load it again.

Note: Suppose you use the first method above. You will notice that there is an Unload button on the SolverTable ribbon. If you click it, the SolverTable.xlam file will close, but the SolverTable item in the add-ins list will still be checked. The effect of this is that if you close Excel and then open it at a later time, SolverTable will reload.

Unloading SolverTable

To unload this add-in (remove it from memory), (1) uncheck the SolverTable box in the Add-ins list (if you used the first method above), (2) close Excel (if you used the second method above), or (3) click on the Unload SolverTable button on the SolverTable ribbon.

An example file

To illustrate SolverTable, I will use the standard product mix model, a version of which appears in Figure 2. (The conventions I use are that the input cells are blue, the changing cells are red, and the objective cell is gray, but you can use any conventions you like.) I assume the Solver dialog box has already been completed in the obvious way (including checking the nonnegative option and specifying the simplex algorithm). The model shown here has already been optimized, but this is not necessary for running SolverTable.

Figure 2

	A	B	C	D	E	F	G	H	I
1	Product mix model						Range names used		
2							Frames_produced	=Model!\$B\$16:\$E\$16	
3	Input data						Maximum_sales	=Model!\$B\$18:\$E\$18	
4	Hourly wage rate	\$8.00					Profit	=Model!\$F\$32	
5	Cost per oz of metal	\$0.50					Resources_available	=Model!\$D\$21:\$D\$23	
6	Cost per oz of glass	\$0.75					Resources_used	=Model!\$B\$21:\$B\$23	
7									
8	Frame type	1	2	3	4				
9	Labor hours per frame	2	1	3	2				
10	Metal (oz.) per frame	4	2	1	2				
11	Glass (oz.) per frame	6	2	1	2				
12	Unit selling price	\$28.50	\$12.50	\$29.25	\$21.50				
13									
14	Production plan								
15	Frame type	1	2	3	4				
16	Frames produced	1000	800	400	0				
17		<=	<=	<=	<=				
18	Maximum sales	1000	2000	500	1000				
19									
20	Resource constraints	Resources used		Resources available					
21	Labor hours	4000	<=	4000					
22	Metal (oz.)	6000	<=	6000					
23	Glass (oz.)	8000	<=	10000					
24									
25	Revenue, cost summary								
26	Frame type	1	2	3	4	Totals			
27	Revenue	\$28,500	\$10,000	\$11,700	\$0	\$50,200			
28	Costs of inputs								
29	Labor	\$16,000	\$6,400	\$9,600	\$0	\$32,000			
30	Metal	\$2,000	\$800	\$200	\$0	\$3,000			
31	Glass	\$4,500	\$1,200	\$300	\$0	\$6,000			
32	Profit	\$6,000	\$1,600	\$1,600	\$0	\$9,200			

Running SolverTable

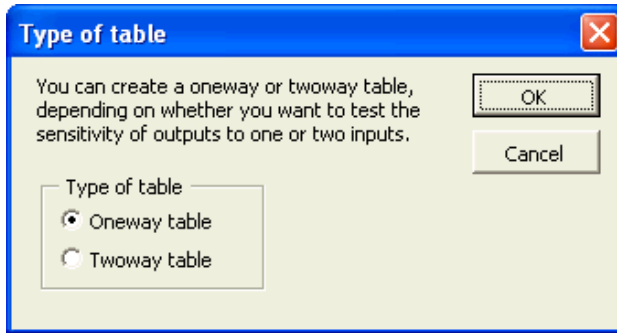
The purpose of SolverTable is very much like that of an Excel data table: to vary one or two inputs and keep track of one or more outputs. The main difference is that SolverTable runs Solver for each input value (or combination of input values) and reports the *optimal* results in the table. Like data tables, there are two options: oneway tables and twoway tables, depending on whether you want to vary one or two *input* cells. However, unlike Excel data tables, there can be multiple output cells for oneway *or* twoway tables. In the latter case, SolverTable creates a separate table for each output cell. Note that a table created by SolverTable retains no links to the original data. If you want to change anything about the table, you must rerun SolverTable.

Creating a Oneway Table

To illustrate a oneway table, I will see how the optimal profit and the optimal product mix (changing cells) vary as the number of labor hours available (cell D21) varies from 2000 to 8000 in increments of 1000. Begin by clicking on the Run SolverTable item on the SolverTable ribbon. You will be asked whether you have a Solver model on this sheet. For this example, you

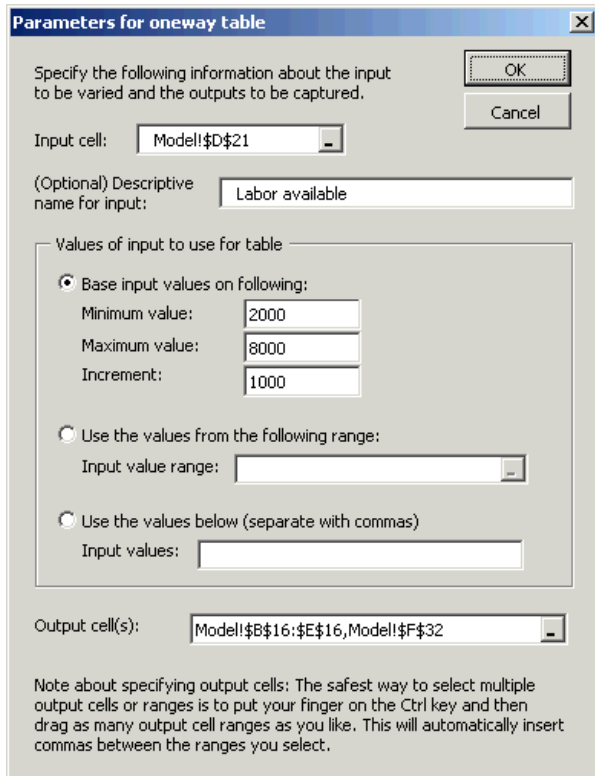
should say Yes. But if you inadvertently try to run SolverTable on a sheet without a Solver model, this gives you a way to back out gracefully. You then see the dialog box in Figure 3. Select the first option and click on OK.

Figure 3



The next dialog box, shown in Figure 4, requires you to specify the input cell, an optional name for the input cell (for labeling the report), the input values (assumed to be in regular increments if you choose the first option, or any list of values if you choose the second or third options), and the output cell(s). SolverTable checks your entries in this dialog box for errors. For example, the input cell should not contain a formula or a label, the output range should not contain the input cell, and so on. However, I cannot guarantee that it checks for *everything*, so be careful when you make these entries. (Check out the note at the bottom of the dialog for selecting multiple ranges.)

Figure 4



The table itself appears in Figure 5. Note that unlike an Excel data table, SolverTable does not put *formulas* at the top of the table (in row 4 of the figure). However, it does put the cell addresses of the output cells. Actually, if the output cells are in named ranges, labels with these names appear in the output rather than cell addresses. For example, the label in cell B4 indicates that this output is the first cell of the Frames_produced range. Similarly, the label in cell F4 indicates that this output is the *only* cell in the Profit range. This way, you can remember which outputs are being reported. Besides these labels, users of previous versions of SolverTable will notice two new features. First, the results of each run are placed on a new worksheet, which are named as STS_n for consecutive values of n. So if you make 3 runs, you will have three new worksheets: STS_1, STS_2, and STS_3. Second, the new worksheet has a built-in chart so that you can graph any of the columns. This is shown in Figure 6. By choosing an output from the dropdown in cell K4, you get a corresponding chart of that column's results. (STS is short for SolverTable Standard, meaning that it is written for the standard version of Solver that ships with Office.)

Figure 5

Creating a Twoway Table

To create a twoway table, go through the same steps as above, but check the twoway option in Figure 3. Then the twoway dialog box in Figure 7 appears. Here, I'll assume that both labor hour availability (D21) and hourly wage rate (B4) are being varied, and that the desired outputs are the number of labor hours used (B21) and the optimal profit. (Any output cells could be selected.)

Figure 7

The dialog box is titled "Parameters for twoway table" and contains the following fields and options:

- Input1 cell: Model!\$D\$21
- Input2 cell: Model!\$B\$4
- (Optional) Descriptive name for input1: Labor available
- (Optional) Descriptive name for input2: Hourly wage
- Values of input1 to use for table:
 - Base input values on following:
 - Minimum value: 2000
 - Maximum value: 8000
 - Increment: 1000
 - Use the values from the following range:
 - Input value range: [empty]
 - Use the values below (separate with commas)
 - Input values: [empty]
- Values of input2 to use for table:
 - Base input values on following:
 - Minimum value: 4
 - Maximum value: 12
 - Increment: 4
 - Use the values from the following range:
 - Input value range: [empty]
 - Use the values below (separate with commas)
 - Input values: [empty]
- Output cell(s): Model!\$B\$21,Model!\$F\$32

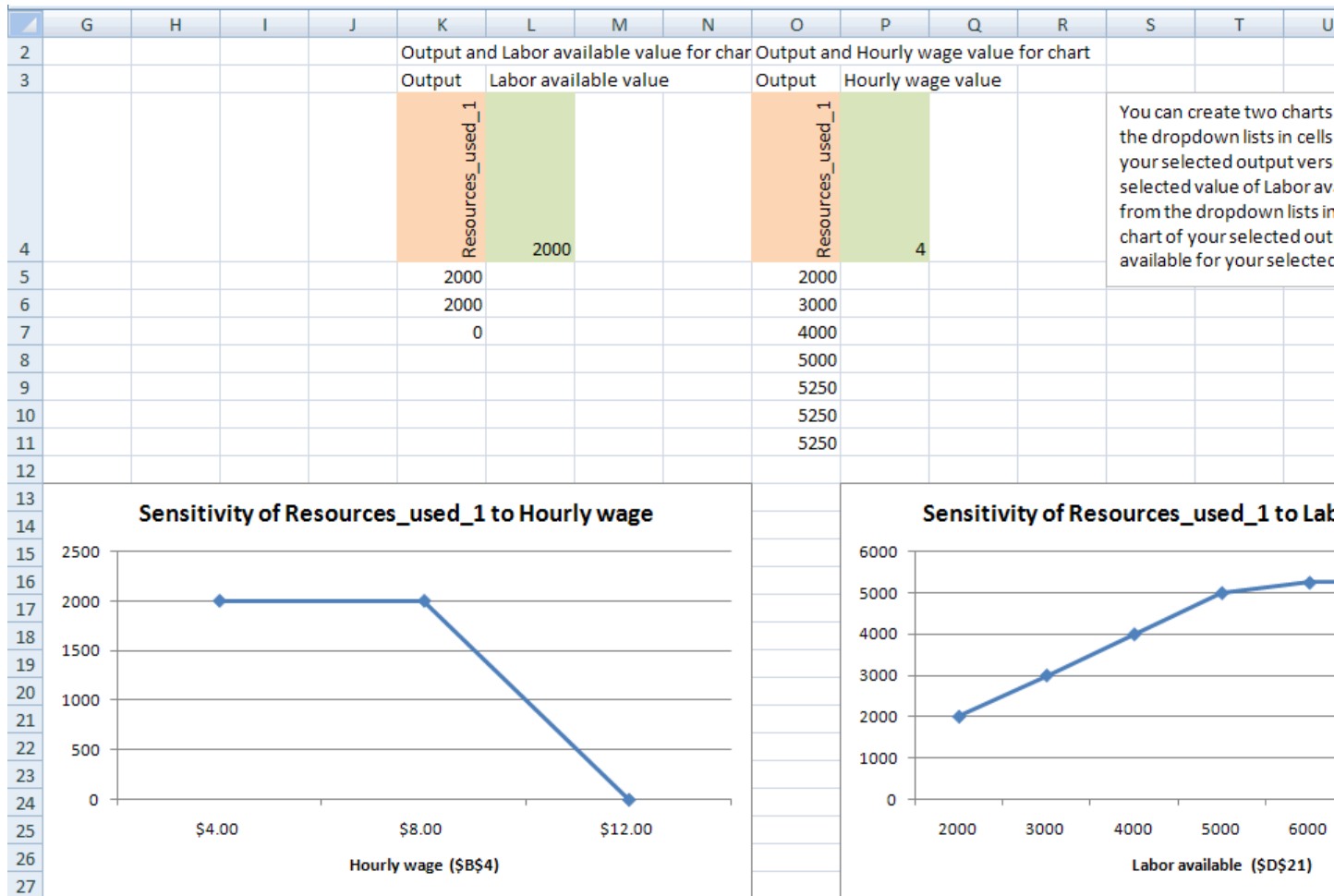
Note about specifying output cells: The safest way to select multiple output cells or ranges is to put your finger on the Ctrl key and then drag as many output cell ranges as you like. This will automatically insert commas between the ranges you select.

SolverTable then creates as many tables as there are output cells, as shown in Figure 8, each with a label for an output cell in the upper left corner. Note that each cell in a table corresponds to a Solver run. For example, when there are 4000 labor hours available and the wage rate is \$4 per hour, the optimal profit is \$25,200 (bottom table) and all 4000 labor hours are used (top table). As before, these results are stored in a new STS_n sheet, and there are now two charts, as shown in Figure 9. As the text box indicates, you can graph any row or any column for any output by choosing from four dropdown lists (the colored cells).

Figure 8

	A	B	C	D	E	F	G	H	I	J
1	Twoway analysis for Solver model in Model worksheet									
2										
3	Labor available (cell \$D\$21) values along side, Hourly wage (cell \$B\$4) values along top, output cell in corner									
4	Resources_used_1	\$4.00	\$8.00	\$12.00						
5	2000	2000	2000	0						
6	3000	3000	3000	0						
7	4000	4000	4000	0						
8	5000	5000	5000	0						
9	6000	5250	5000	0						
10	7000	5250	5000	0						
11	8000	5250	5000	0						
12										
13	Profit	\$4.00	\$8.00	\$12.00						
14	2000	\$14,000	\$6,000	\$0						
15	3000	\$20,000	\$8,000	\$0						
16	4000	\$25,200	\$9,200	\$0						
17	5000	\$30,250	\$10,250	\$0						
18	6000	\$31,250	\$10,250	\$0						
19	7000	\$31,250	\$10,250	\$0						
20	8000	\$31,250	\$10,250	\$0						

Figure 9



As with all Solver models, it is a good idea to take a close look at the results, numerical and graphical, and see whether they make sense. For example, column D in Figure 8 contains all 0's. Does this make sense? Yes, it probably does. When the labor rate is \$12 per hour and all other monetary inputs stay constant, labor is evidently too expensive for the company to make a profit. So it produces nothing!

Creative Uses of SolverTable

With some imagination, you can get SolverTable to run some really interesting sensitivity analyses. For example, suppose you want to allow the availabilities of all three resources (labor, metal, and glass) to change by a common factor. Then you need to change the model slightly, as indicated in Figure 10. Now the original availabilities have been moved to column F, a change factor has been inserted in column H, and *formulas* have been entered for availabilities in column D. Specifically, the formula in cell D21 is =F21*\$H\$21, which is then copied down. (**Note:** You should *type* the original values in column F. You should *not* copy the original values from column D to column F and *then* enter formulas in column D. The reason is that Solver will then think the right sides of the constraints are in column F, not D, and the values in column F will *not* change in your SolverTable run.)

Figure 10

	A	B	C	D	E	F	G	H
20	Resource constraints	Resources used		Resources available		Original values		Change factor
21	Labor hours	4000	<=	4000		4000		1.0
22	Metal (oz.)	6000	<=	6000		6000		
23	Glass (oz.)	8000	<=	10000		10000		

You can now do a oneway sensitivity analysis on this change factor, using the settings in Figure 11. The corresponding results appear in Figure 12. By the way, if your numbers in column A aren't formatted very nicely (too many decimals, say), be aware that SolverTable formats the various input values in the table the same as the input cell itself (in this case, cell H21). So it's a good idea to format input cells the way you want them *before* running SolverTable.

Figure 11

Parameters for oneway table

Specify the following information about the input to be varied and the outputs to be captured.

Input cell: Model!\$H\$21

(Optional) Descriptive name for input: Change factor

Values of input to use for table

Base input values on following:

Minimum value: .6

Maximum value: 1.4

Increment: .1

Use the values from the following range:

Input value range:

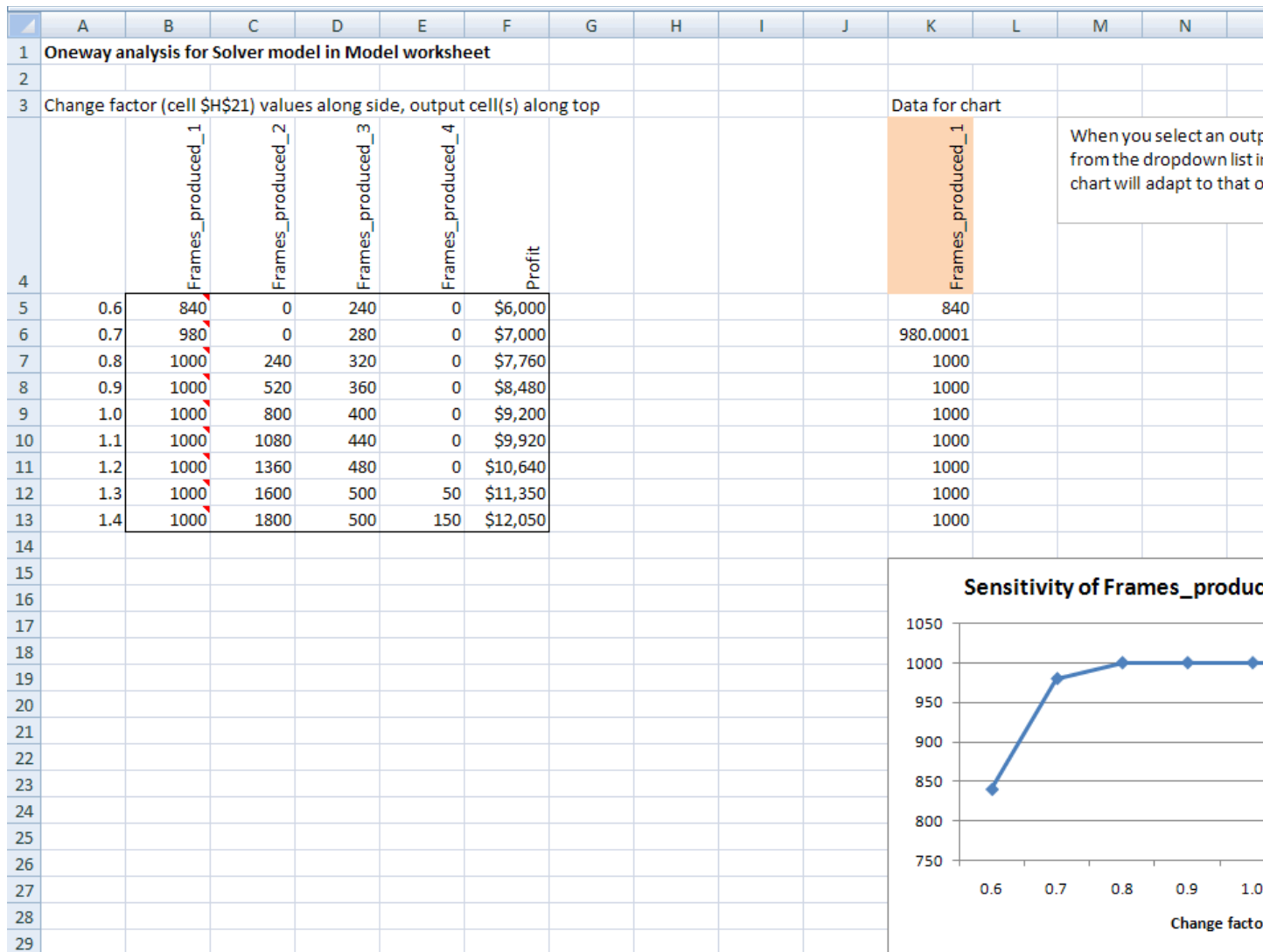
Use the values below (separate with commas)

Input values:

Output cell(s): Model!\$B\$16:\$E\$16,Model!\$F\$32

Note about specifying output cells: The safest way to select multiple output cells or ranges is to put your finger on the Ctrl key and then drag as many output cell ranges as you like. This will automatically insert commas between the ranges you select.

Figure 12



As another example, suppose you want to keep track of a *function* of several cells in the model. For example, suppose you want to keep track of the maximum number of frames of any single type produced, that is, the maximum of the changing cell values. You can't select this as an output *directly* because it doesn't appear anywhere in the model. But there is a simple solution: Create a formula for it in some unused cell, and then specify this cell as an output cell.

Other Notes

- If you run a oneway table for a particular model and then run another, the settings dialog box (see Figure 4 or 11 above) show the *previous* settings as a starting point. Of course, you can change any or all of these. The same is true if you have run a twoway table for a particular model. (In case you are interested, these settings are stored in a new worksheet called something like **Model_STS** that is inserted into your workbook. (It has the name of the model sheet followed by “_STS.”) However, this worksheet is “very hidden”—you can't get to it except by writing VBA code. If you really want to look at this hidden sheet,

or even delete it, you can do so, but it takes a bit of work. First, press Alt-F11 to get to the Visual Basic Editor. On the left side of the screen, in the Project Explorer, select any of the items in the file your model is in. If you have run SolverTable on it, you should see that one item in the file is the hidden worksheet. Now press Ctrl-g to open the Immediate Window, where you can run single lines of VBA code. In the Immediate Window, type the line **Worksheets("Model_STS").Visible = True** (or whatever the name of the hidden sheet is) and then press Enter. Now the hidden sheet is no longer hidden, and you can do anything you want with it.)

- When you run SolverTable, the active sheet should be the sheet the model is on. If you forget and run SolverTable when some other sheet is active, you can always cancel out of SolverTable and activate the model sheet.
- As programmers realize, it is practically impossible to catch all of the bugs. The current version of SolverTable tries to “trap” any error it encounters and present a message in a dialog box that identifies the error. If you see one of these messages, please report its information to me, so that I can see what error you encountered.

Chris Albright, Kelley School of Business, Indiana University, albright@indiana.edu,
November 2009