Taylor Lovell
Chart Builder

Executive Summary

Utah Valley University (UVU) is an institution of higher education and is the largest public university in the state. The department for which I work specifically, Institutional Research and Information, is responsible for all surveys administered to students, employees, alumni, and others.

One such survey is entitled “Great Colleges to Work For,” a national survey of full-time employees in higher education. This survey is conducted externally by a company called ModernThink. It is comprised of 60 statements about working at UVU, to which employees respond on a 5-point agreement scale (“Strongly Agree” to “Strongly Disagree”). Because of quirks in the way the data are presented, the reports delivered by ModernThink are almost unusable for our purposes.

In order to remedy this situation, the Graph Builder workbook contains a macro that transfers all of the data from this report to an Excel sheet. Below is a brief overview of the major steps taken by this macro:

1. Import data from a text file
2. Extract all important data from the text
3. Format the data into tables
4. Produce a chart for each table

After running this macro, the user will have an easy-to-read Excel sheet containing all 60 statements and a chart for each. From this sheet it will be much easier to produce a usable and valuable report.

Contents

Implementation

The “Great Colleges to Work For” survey report is delivered by ModernThink in PDF format. The charts in this report have a tendency of misrepresenting data for the sake of aesthetics, and are therefore of questionable integrity. The actual data from the survey is very expensive to purchase, but it is possible to extract the necessary data from the report. This program does just that, as well as produces accurate and valuable tables.

Buttons

The first thing that the user sees when he opens the Graph Builder workbook is three buttons (see Figure 1). Each button has a specific purpose discussed below, and each button remains on the sheet after the macro has been run.

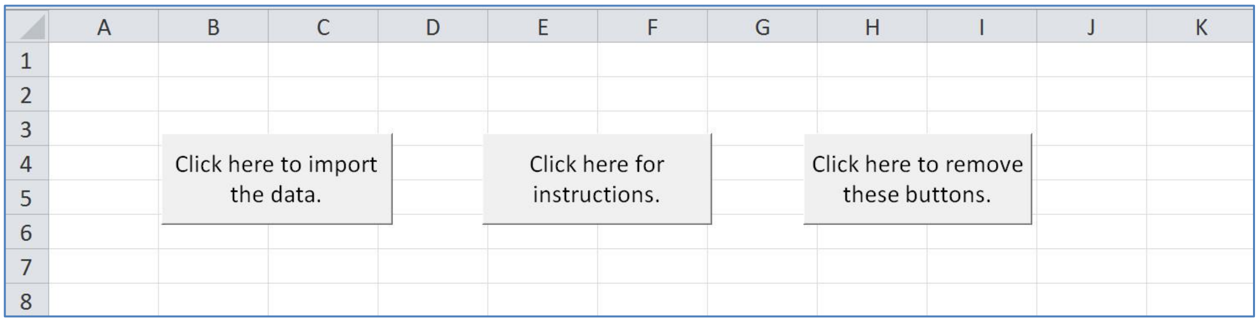
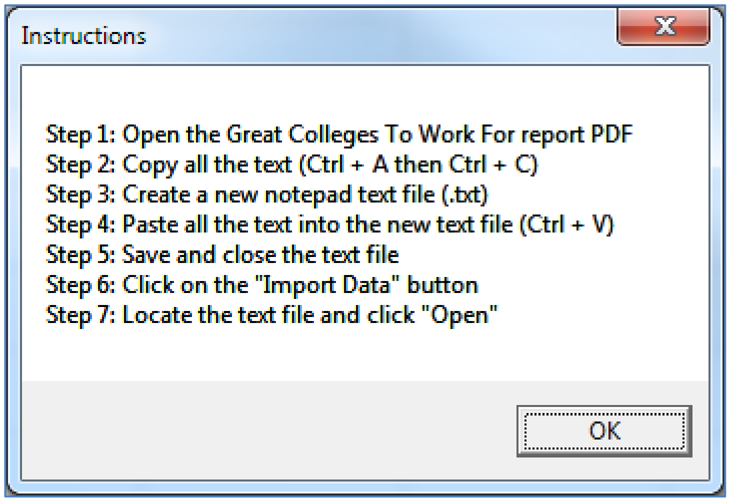

A
B
C
D
E
F
G
H
I
J
K
1
2
3
4
5
6
7
8
Click here to import the data.
Click here for instructions.
Click here to remove these buttons.

Figure 1

“Click here to import data.” Clicking this button will execute the *startHere* macro, which imports all of the data, formats all of the tables, and produces all of the charts.

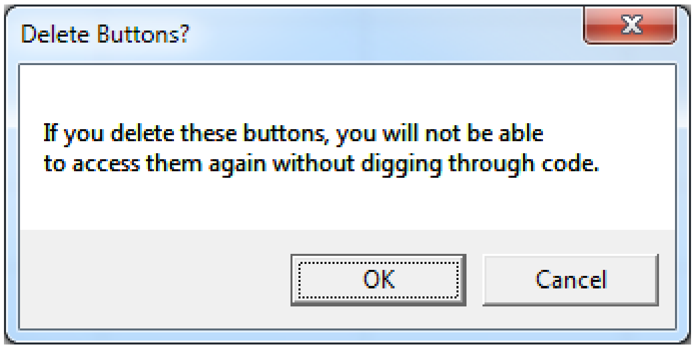
“Click here for instructions.” Clicking this button will display the instructions (see Figure 2) that the user must take in order for the *startHere* macro to be successful.



Step 1: Open the Great Colleges To Work For report PDF
Step 2: Copy all the text (Ctrl + A then Ctrl + C)
Step 3: Create a new notepad text file (.txt)
Step 4: Paste all the text into the new text file (Ctrl + V)
Step 5: Save and close the text file
Step 6: Click on the "Import Data" button
Step 7: Locate the text file and click "Open"

Figure 2

“Click here to remove these buttons.” Once the user is happy with the report, clicking this button will remove all buttons from the sheet. The user is warned that the buttons will be difficult to bring back (see Figure 3). If the user clicks OK, all buttons are deleted.


Delete Buttons?

If you delete these buttons, you will not be able
to access them again without digging through code.

Figure 3

Running the Macro

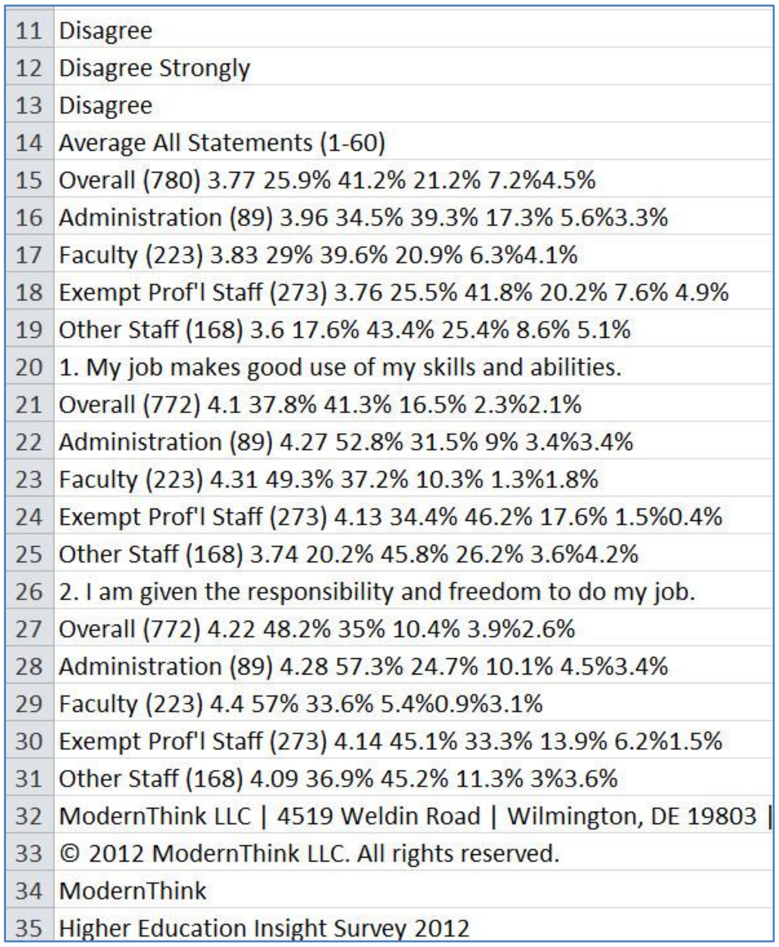
As stated above, the *startHere* macro is responsible for all parts of this program: importing the data, creating tables, etc. The code for each of these operations does not reside within *startHere* itself, but rather in sub procedures which are called by *startHere*. Each of these sub procedures is described below in the order in which they are called.

importData. This is the first sub procedure called by *startHere*. First it creates a FileDialog object and sets it as the type msoFileDialogFilePicker. The file dialog box is then opened and the user is asked to select the text file containing the Great Colleges data (see “Click here for instructions.” above). If the user didn’t select a file or if he selected a non-text file, the program ends and provides the user with the appropriate warning. *importData* then adds a blank sheet to the current workbook, names it “Imported Data,” and deletes all other sheets. Finally, *importData* imports the data into the “Imported Data” sheet:

```
ActiveSheet.QueryTables.Add(Connection:="TEXT;" & fd.SelectedItems(1),
    Destination:= Range("A1")).Refresh
```

addSheet. This is a simple sub procedure which adds a new blank sheet and names it “Formatted Data.”

Figure 4 below shows how the imported data is presented. In this figure, statements are presented on rows 14, 20, and 26. After each statement is a breakdown of responses by each of the four job categories (administration, faculty, exempt staff, and non-exempt staff) as well as the average of all four. Contained within the parentheses is the number of respondents for each category. The number immediately after the parentheses is the average score for each category on the specific statements. This is calculated by awarding points for each response (e.g. "Strongly Agree" = 5, "Strongly Disagree" = 5) and determining the average. The final numbers are percentages for the individual categories.



11	Disagree
12	Disagree Strongly
13	Disagree
14	Average All Statements (1-60)
15	Overall (780) 3.77 25.9% 41.2% 21.2% 7.2%4.5%
16	Administration (89) 3.96 34.5% 39.3% 17.3% 5.6%3.3%
17	Faculty (223) 3.83 29% 39.6% 20.9% 6.3%4.1%
18	Exempt Prof'l Staff (273) 3.76 25.5% 41.8% 20.2% 7.6% 4.9%
19	Other Staff (168) 3.6 17.6% 43.4% 25.4% 8.6% 5.1%
20	1. My job makes good use of my skills and abilities.
21	Overall (772) 4.1 37.8% 41.3% 16.5% 2.3%2.1%
22	Administration (89) 4.27 52.8% 31.5% 9% 3.4%3.4%
23	Faculty (223) 4.31 49.3% 37.2% 10.3% 1.3%1.8%
24	Exempt Prof'l Staff (273) 4.13 34.4% 46.2% 17.6% 1.5%0.4%
25	Other Staff (168) 3.74 20.2% 45.8% 26.2% 3.6%4.2%
26	2. I am given the responsibility and freedom to do my job.
27	Overall (772) 4.22 48.2% 35% 10.4% 3.9%2.6%
28	Administration (89) 4.28 57.3% 24.7% 10.1% 4.5%3.4%
29	Faculty (223) 4.4 57% 33.6% 5.4%0.9%3.1%
30	Exempt Prof'l Staff (273) 4.14 45.1% 33.3% 13.9% 6.2%1.5%
31	Other Staff (168) 4.09 36.9% 45.2% 11.3% 3%3.6%
32	ModernThink LLC | 4519 Weldin Road | Wilmington, DE 19803 |
33	© 2012 ModernThink LLC. All rights reserved.
34	ModernThink
35	Higher Education Insight Survey 2012

Figure 4

At this stage, the *startHere* macro looks for the "Average All Statements (1-60)" cell in the Imported Data sheet, activates it, and runs the next sub procedure.

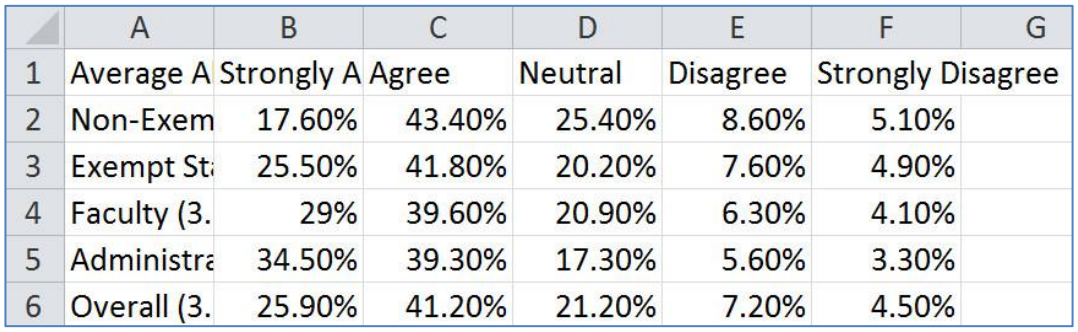
***transferData*.** This sub procedure first creates a place-holder range on the Formatted Data table: the cell into which the whole statement is copied. Labels are then copied into the cells right of the placeholder cell for each of the agreement categories (e.g. "Strongly Agree"). The remainder of this sub procedure occurs five times within a For loop, once for each job category. *transferData* determines the job category, extracts the average score, and pastes this information into the appropriate cell in the Formatted Data sheet. The sub procedure on the next page is then called.

recordScores. The first step is to extract just the percentages. The program locates the end parenthesis and skips past the average score, copying the remainder of the cell to a string variable. Then, using two “cursor” variables, *recordScores* locates the next percent symbol and copies each individual data point into the appropriate variable. After recording the “Strongly Agree” data, the following is repeated for each additional level of agreement with cursor1 and cursor2 swapping places:

```
cursor2 = InStr(cursor1 + 1, dataString, "%") 'stores the position of the 2nd percent sign
agree = Mid(dataString, cursor1 + 1, cursor2 - cursor1) 'extracts the second value
```

Once all the data have been stored in the variables, they are copied to the appropriate cells in the Formatted Data sheet.

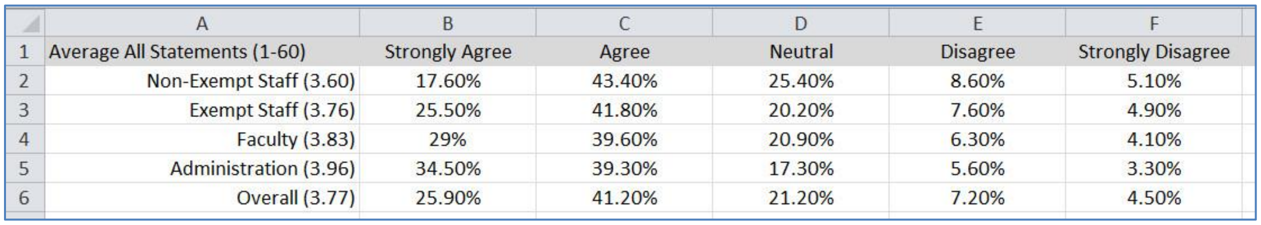
Now that the data for “Average All Statements (1-60)” has been transferred, the same is done for all 60 statements. *startHere* finds the next statement by checking whether each cell starts with a number, activating the next if it does not. Figure 5 shows what the data looks like once it has been transferred.



	A	B	C	D	E	F	G
1	Average A	Strongly A	Agree	Neutral	Disagree	Strongly Disagree	
2	Non-Exem	17.60%	43.40%	25.40%	8.60%	5.10%	
3	Exempt St	25.50%	41.80%	20.20%	7.60%	4.90%	
4	Faculty (3.	29%	39.60%	20.90%	6.30%	4.10%	
5	Administra	34.50%	39.30%	17.30%	5.60%	3.30%	
6	Overall (3.	25.90%	41.20%	21.20%	7.20%	4.50%	

Figure 5

formatTables. Now that all the data have been transferred, this sub procedure formats it to look better for the user. It sets the alignment and width for all columns and adds a grey highlight to the rows containing the individual statements. Figure 6 shows what the formatted data looks like.



	A	B	C	D	E	F
1	Average All Statements (1-60)	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
2	Non-Exempt Staff (3.60)	17.60%	43.40%	25.40%	8.60%	5.10%
3	Exempt Staff (3.76)	25.50%	41.80%	20.20%	7.60%	4.90%
4	Faculty (3.83)	29%	39.60%	20.90%	6.30%	4.10%
5	Administration (3.96)	34.50%	39.30%	17.30%	5.60%	3.30%
6	Overall (3.77)	25.90%	41.20%	21.20%	7.20%	4.50%

Figure 6

createCharts. The main purpose of this program is the creation of charts for transfer to another report. That is the purpose of this sub procedure. *createCharts* takes the following steps:

1. Select all of the data for one statement
2. Add the chart of type 100% Stacked Bar
3. Change the font to Arial and size 11
4. Add the statement as the chart's title
5. Set the chart title font to size 14
6. Swap the chart's row column data
7. Move the legend to the top of the chart
8. Adjust the width of the bars in the chart
9. Add data labels
10. Move the chart next to its table
11. Shrink the chart so all will fit
12. Select colors for the bars

Figure 7 shows what the charts look like after they have been created.

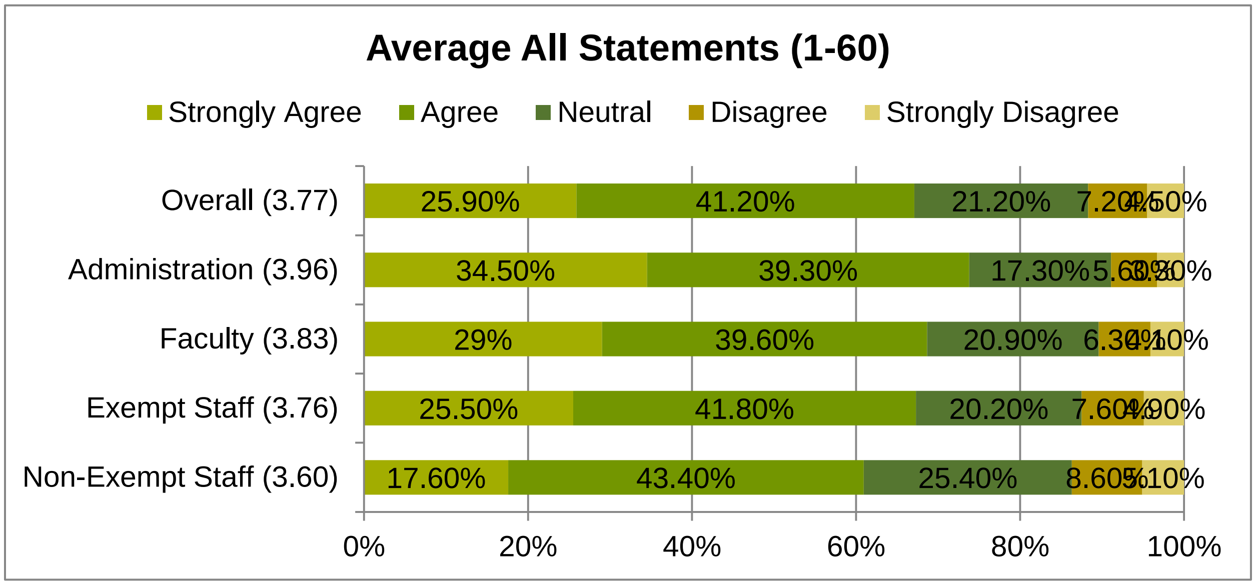


Figure 7

addButtons. This is the final sub procedure of the program. It adds to the Formatted Data sheet the same buttons which the user was presented with at the start. This allows the user to refresh the data if he so chooses.

Taylor Lovell
Chart Builder

Troubleshooting

When I first started this project, my main concerns were (1) extracting the percentages from each line and (2) creating the charts.

Extracting Percentages. Because of the way that the data is presented in the PDF report, there is not a uniform way that the data is copied and pasted. Sometimes there will be a space between percentages, sometimes there won't be. The number of digits ranges from one to four, without much indicator of such. To overcome this challenge, I used two "cursor" variables to keep track of where percentage signs were, and then simply took all the text in between them. It may not be the most elegant solution, but I am pleased with it.

Creating the Charts. I knew there was going to be a lot of formatting that I had to do to make the charts look the way I wanted them to look. Fortunately, the Record Macro feature of VBA proved most useful. I was able to quickly see how to make each change that I wanted, and implement it into the code. The only real hiccup came when it was time to add the title. The code which was produced by the recorder did in fact add a title, but when ran with all other code it produced an error. I had to search the internet for an alternative line of code to add the title, which proved successful.

Additional Troubleshooting. One unforeseen quirk that I encountered was when a statement ran onto two lines. Since my code relies so much on the active cell, this created misplaced data. I tried offsetting the active cell by one, but this ruined data for the rest of the program. To solve the problem, I set the active cell like normal, but when it came time to transfer the statement and concatenated both lines.

This has been a great project to not only help streamline an otherwise time consuming process at work, but to put together everything that I have learned this semester. I was able to implement tools which I already possessed and learn new ones that I had never before used. There is plenty of VBA that I do not know, but I Feel confident in my abilities to discover how to do most things that I will want.