Cost of Living Analyzer

<u>Executive Summary</u>

This workbook was designed to make it easier for a person (specifically a business student) to evaluate a job offer. The biggest issue with this is gathering enough data on the offer location and comparing it to a known location. While this can be achieved with multiple inquiries to a website and recording the data, that process can be tedious and time consuming. This workbook automates the process and gives the user an easy to read, onscreen report of the evaluation. The result is a concise report can be used to assist the user assess the true value of the offer, and whether or not the offer and location meet the needs of the user. Ultimately the user needs to make the decision; this is a tool to help make an educated decision.

The report has several key areas: Financial, Benefits, Education, Housing, Cost of Living, and Crime. The Financial section displays user inputs for Salary, Vacation, and Relocation. The program common sizes the salaries and bonuses using the Cost of Living factor. The Benefits section simply displays user inputs to allow the user to verify benefits packages. The right hand side of the report displays critical statistics for Education, Housing, Cost of Living, and Crime from both locations and national rates/averages.

The desired result of this project is to help user's make informed decisions in an efficient manner. Through the automation of the web query and data compilation, the program will provide an extensive, customized analysis in the time that would normally be required to simply obtain cost of living data from a website.

The purpose of the workbook is to allow the user to enter easily compare their current (our previous) job and location to a job offer and its respective location. Once information has been entered, the user will get an on screen report, and have the opportunity to print the report. This workbook is designed to help the user make an informed decision about an offer by comparing it to a location and income level they are familiar with, by comparing both amounts against National Averages for various expenses. This form could also be used to compare two offer locations by entering one of the offers in place of the base location, but this removes the frame of reference from the results.
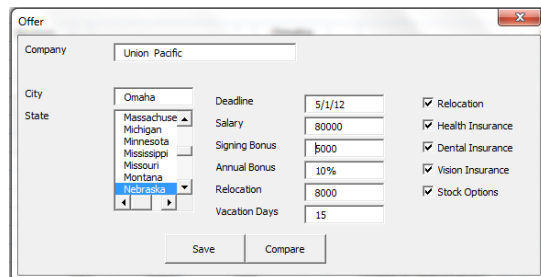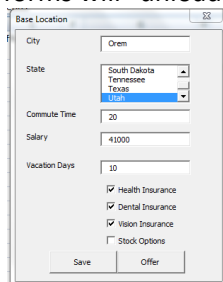
The workbook consists of 4 sheets, "Offer", "Results", "States", and "Webquery":
- The Offer sheet is used to store user inputs as well as receive information from the web query.
- The Results sheet is the basis for the printed report with cells formatted and receives information from frmresult.
- The States sheet is simply a database of state names that is used to populate the states listboxes on frmBaseLocation and frmOffer and therefore is hidden from view.
- The Webquery sheet is used by the web queries to pull data from the web then data is transferred to the Offer sheet.
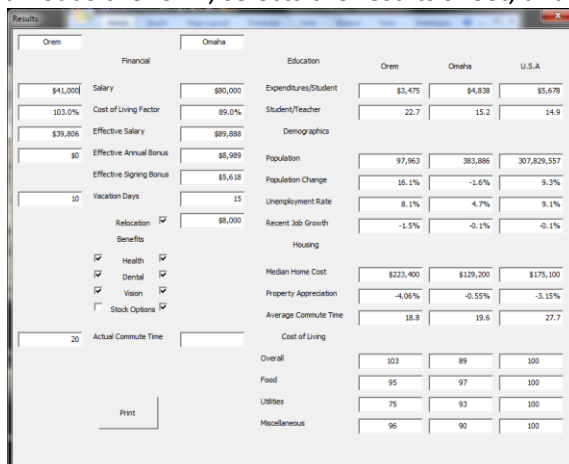
The user only interacts with the "Results" sheet, and that is only for printing.

There are 3 userforms, frmBaseLocation, frmOffer, and frmResults:
- frmBaseLocation and frmOffer uses textboxes, checkboxes, and a listbox to request information from the user, the city and state must be filled in for the web query to execute. The form initializes with information from rows 3 and 4 of the "Offer" sheet, and overwrites the rows when the save or offer/compare buttons are selected. This was programmed with textbox = statements, a do loop for the state list box, and if instr statements to populate the checkboxes, with saving programmed in reverse fashion. When the Offer or Compare buttons are used, the forms will "unload", allowing the user to flow from form to form.

- frmResults initializes with the textboxes filled in by the data stored on rows 2-4 of the Offer sheet. The print button is one that I'm proud of, even though it's very simple. The print button unloads the form, selects the results sheet, and previews it.
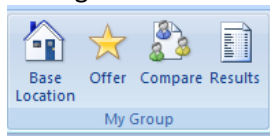
- There is a frmCompare that was going to be used to allow the user to select comparisons to run, but this was deactivated since most of the web pages were already being for the basic data. Since all of the data fit on a single page, I felt it was acceptable to provide additional data.
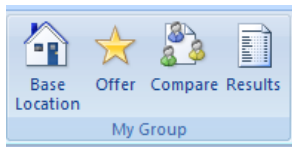
The workbook has two modules, CompareMod and Ribbon:
- CompareMod runs multiple web queries to obtain data on each location, and it copies and pastes necessary data to the Offer sheet. The data collecting navigates via the offset function, pulling the national numbers at the same time it pulls the base location, so the national numbers refresh each time the query is run. I felt this was important since the numbers do change. This module was designed so that the form could be expanded relatively easily to bring in even more data from the web, if the user desired.
- Ribbon controls the custom buttons to initial the various forms and queries. I designed the program so that the user could save a base location, then simply enter new offers or modify the existing offer.

User's Guide

The user will navigate the program via the "My Group" icons in the upper right hand corner of the ribbon.

The user will initiate the program by selecting the Base Location button located on the right hand side of the header. The Base Location user form will now display.

The Base Location form requests basic information for the user that will serve as a reference point for the comparison. The user should enter their current city or the city where they most recently working (in the case of students), along with their income. This form only requires city and state information, but the more information entered, the better the comparison will be. Once all applicable fields are entered, the user will click the save button, or the offer button. Both buttons will save the data,

allowing it to be accessed the next time the form is initialized. The save button will then close the form. The offer button will also close the Base Location form and display the offer form.

The next step is to enter your job offer information. The offer form can be initialized by clicking the offer button on the Base Location form, or by selecting the offer button on the right hand side of the header. With the offer form displayed, the user will need to enter all information from the offer. Again the only required fields are the city and state, but the more information entered, the more useful the results will be. The save button will save the data entered, allowing it to be accessed again the next time the form is initialized, and close the form. The compare button will also save the data and initialize the web query that will retrieve information about both the base location and the offer location. Once the information has been retrieved, the results form will display.

The results form gives an onscreen display of the comparison. While the user form itself cannot be printed, there is a print button that will allow the user to get a full printout of the form details. To print the results, the user will simply click the print button on the bottom right side of the results form. This button will load the results to the results sheet in the workbook, allowing the user to access results at a later time, and it will hide the results form and display a print preview that the user can then print. The worksheet can also be printed through the Excel print function. In addition to the two locations entered, the far right column of the results displays National statistics for applicable criteria.

Union Pacific offer in Omaha, Nebraska expiring on 5/1/12

Orem	Financial	Omaha	Education	Orem	Omaha	U.S.A
$41,000	Salary	$80,000	Expenditures/Student	$3,475	$4,838	$5,678
103.0%	Cost of Living Factor	89.0%	Student/Teacher	22.7	15.2	14.9
$39,806	Effective Salary	$89,888				
$0	Effective Annual Bonus	$8,989	Demographics			
	Effective Signing Bonus	$5,618	Population	97,963	383,886	307,829,557
10	Vacation Days	15	Population Change	16.1%	-1.6%	9.3%
	Relocation	$8,000	Unemployment Rate	8.1%	4.7%	9.1%
			Recent Job Growth	-1.5%	-0.1%	-0.1%

Benefits

				Housing			
X	Health	X		Median Home Cost	$223,400	$129,200	$175,100
X	Dental	X		Property Appreciation	-4.06%	-0.55%	-3.15%
X	Vision	X		Average Commute Time	18.8	19.6	27.7
	Stock Options	X					

20	Actual Commute Time		Cost of Living			
			Overall	103	89	100
			Food	95	97	100
			Utilities	75	93	100
			Miscellaneous	96	90	100

Learning and Conceptual Difficulties

For this project, I dove heavily into userforms. Getting the forms to be functional and flow logically took several revisions, but in the end I was to come up with forms that met my needs. I learned a lot about what it takes to make quality userforms (I think).

The web query caused two main challenges. I had intended to use the bestplaces.net's compare function to simply pull CoL numbers, however the comparison url uses a numeric code for each city (not a zip code). My next thought was to use just the bestplaces.net/city/"state"/"city" for the query. This provides a simple overview, with basic data that didn't really meet the original scope of the project, and would have to be run for each location. This also required pulling numbers out of the middle of lines of text which would have taken more coding to get less data. My final solution was to run 10 queries to bestplaces.net/"category"/city/"state"/"city". This provides significantly more data, but requires slightly more processing time.   This allowed the project to creep, but also allowed for more comprehensive results.

The other issue that I encountered was with printing the results. I initially thought it would be possible to print the userform directly, but I learned that this is not possible. To resolve this, I created the "Results" sheet which I formatted so that the report would be aesthetically pleasing. Had I known this in the first place, I would have simply used the "Results" sheet instead of the "Offer" sheet – both sheets end up with the same data, just formatted differently. This was a good learning experience; I know how to layout my data next time to reduce duplications.

Assistance

Though I discussed several concepts with Prof. Allen, James Hall, David Tate, and David Long, I am comfortable saying that I wrote all of the code in this program myself. I also referred to several projects from earlier in the semester to help with coding when I ran into problems.