

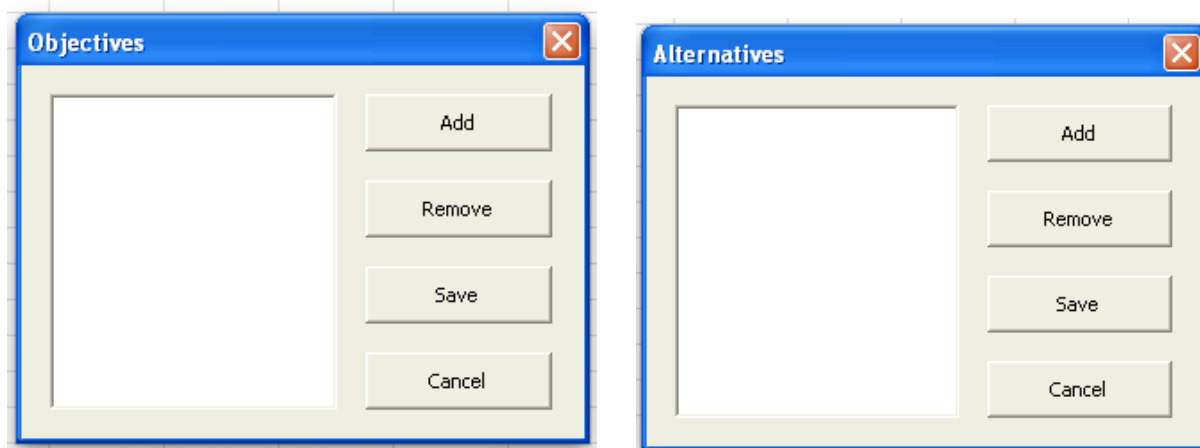
## Final Project Overview: Analytic Hierarchy Process Automation

### Executive Summary

The first semester of my graduate program in Public Administration I took a Decision Analysis class that used Excel as it's primary tool to teach us how to make informed future decisions using what is called the Analytic Hierarchy model. The model involves entering in the alternatives and objectives you have for a decision, then comparing both the alternatives to each other and the objectives to each other to see why objectives are more important and which alternatives are more desirable when compared. With that information and through a bunch of formulas, a summary page is created that shows which alternative wins most often in a series of trials. It is rather tedious to recreate all of these formulas and tables each time you want to make a decision, so I created a VBA project that automates the process. Using a series of buttons, user forms, and input boxes, I take you through the process of entering in the pertinent data and then Excel automatically crunches the numbers and comes up with a summary page.

### Implementation Documentation

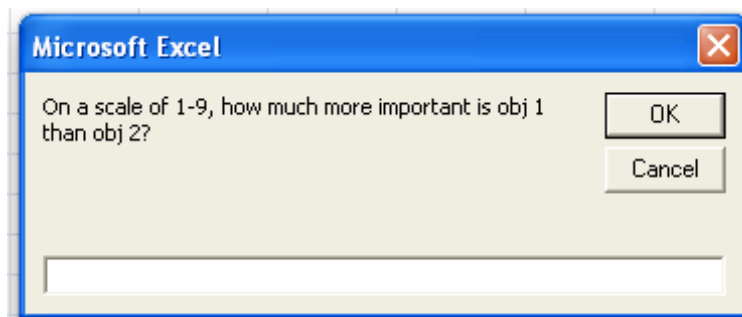
***Alternatives and Objectives:*** The first sheet in the Excel spreadsheet is called Buttons. It contains the two buttons needed to enter in your alternatives and objectives for the decision you are trying to make. When the buttons are pushed, these user forms pop up:



The image displays two side-by-side user forms. The left form is titled 'Objectives' and the right form is titled 'Alternatives'. Both forms have a blue title bar with a close button (X) in the top right corner. Each form contains a large, empty text input area on the left side. To the right of the input area, there are four buttons stacked vertically: 'Add', 'Remove', 'Save', and 'Cancel'. The buttons are light gray with black text.

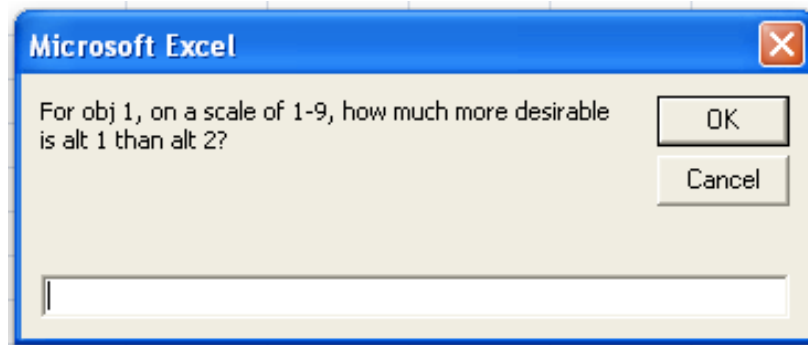
The user is free to add objectives and alternatives using these forms. If they type them in incorrectly, they can highlight and remove them as necessary. When the user hits save, those objectives and alternatives are saved into the summary sheet, and then into arrays that hold the values. Two variables are set with the number of objectives and alternatives the user enters in. This is really the beauty of the whole project. The project is put together in such a way that the user can enter as few or as many alternatives and objectives as they would like. The program is dynamic and adjusts accordingly.

**Objectives sheet:** When the user clicks on the objectives sheet, they see a button that says Objectives pairwise. When they click it, a whole series of things happens. A table with the objectives on the row and the column is created and an input box pops up that runs the user through a series of questions comparing the objectives to each other. The question is, “On a scale of 1-9, how much more important is [objective] than [objective]”? The prompts compare all the objectives to each other and fills in the slots in the table as it receives the user’s answers. When all the spots are filled, the input box goes away and the sheet populates itself with a table that normalizes the values entered in the objectives pairwise table, determines the average of each row in that table, the standard deviation, then uses these numbers to determine the norm inverse of each row, and finally normalizes these values. These numbers are used as weights in the final decision. If the user enters in the values in their pairwise comparison incorrectly, they can press the Objectives Pairwise button again, but at this point they will have to enter all the values in again. At some future point I will hopefully find a way to change just one value automatically, but at the moment, the user will just need to type it into the actual table if they make a mistake, after the prompts are over.



A Microsoft Excel dialog box with a blue title bar and a close button (X) in the top right corner. The text inside the dialog box reads: "On a scale of 1-9, how much more important is obj 1 than obj 2?". To the right of the text are two buttons: "OK" and "Cancel". Below the text is a large, empty white input field.

**Alternatives sheet:** Once the user is done with the objectives, they click on the Alternatives sheet. This sheet also has a button on it, this one labeled Alternatives Pairwise. When they click that button, the first thing they see is pairwise tables similar to the one in the Objectives sheet, but these have the Alternatives listed on each column and row and in the corner of each table is the name of one of the objectives. There are pairwise tables for each objective entered in the original Buttons sheet. Another series of input box prompts pop up to guide the user through entering their values into each of the pairwise tables. The boxes are customized to ask the user which alternative they feel is more desirable, based on each of the objectives. An example of these prompts looks like this:



A Microsoft Excel dialog box with a blue title bar and a close button (X) in the top right corner. The text inside the dialog box reads: "For obj 1, on a scale of 1-9, how much more desirable is alt 1 than alt 2?". To the right of the text are two buttons: "OK" and "Cancel". Below the text is a large, empty white input field.

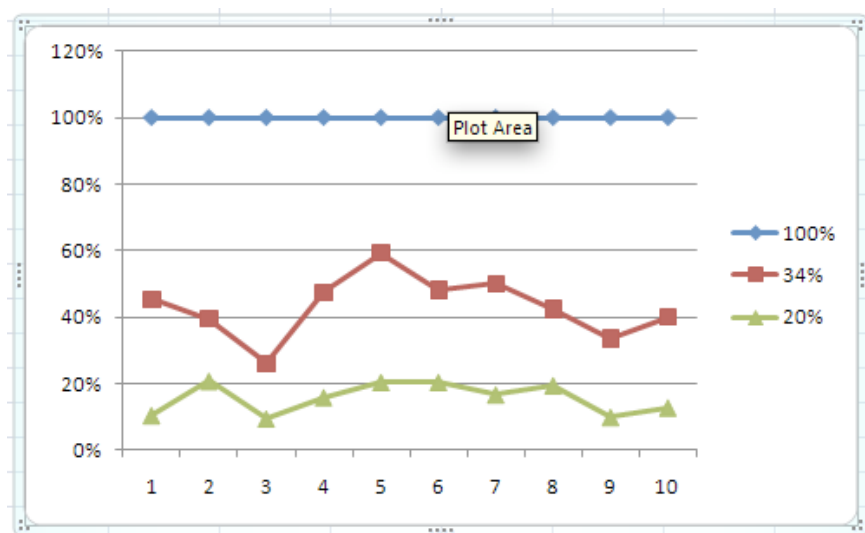
Once the user goes through all of these prompts, the same tables and data as in the Objectives sheet are populated—a normalized table of the pairwise values, average, standard deviation, norm

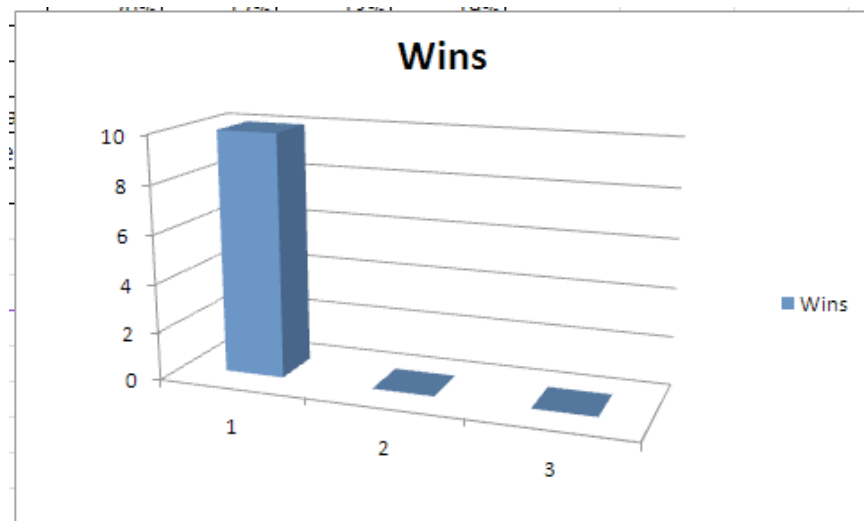
inverse and a normalized norm inverse. These values are used in the summary sheet in the final consequence table, which is part of the summary of the decision making process.

**Summary sheet:** Once all the values have been entered in the Alternatives sheet, the user can then click on the summary sheet and press the button called Calculate Summary. Another series of things will happen all in a row. A consequence table will populate itself. This shows the objectives on the column and the alternatives across the top row of the table. The values in the table are the normalized values pulled from the Alternatives sheet. The weights are the normalized values from the Objectives sheet. This table essentially gives the user values on a percentage scale based on each objective of how desirable they feel each alternative is. The weights are how important the user feels each objective is.

Several formulas are then calculated below this consequence table. First is the Weighted Total. This formula is a SUMPRODUCT of each alternative column with the weights. It shows which alternative is most desirable in a single number. The next row has the Percent of Best. This is the same thing as the Weighted Total, except it normalizes it so that whichever alternative is best is rated 100% and the others are some lesser number. The third row is the Rank. Again, it is the same as the above two, just in number form.

Once these formulas are calculated, a data table is created with 10 iterations of the Percent of Best formula line. A detail I have not mentioned to this point is that this Analytic Hierarchy Process builds uncertainty into the values the user enters in for the pairwise tables. By using the RAND() formula in the NORMINV() formula, the values become “fuzzy,” as my professor refers to them. Essentially they use the properties of a bell curve and the values the user enters in to show the variability in the decision process based on the user’s standard deviation. So when I say the data table has ten iterations of the Percent of Best formula, it runs the formula ten times and each time the numbers change based on the NORMINV and STDEV formulas used earlier. Once this table is created, two graphs are created. One is called a dancing line graph and it is a line graph of the results of the data table, which if you press F9 repeatedly, it will “dance.” It’s a great representation of the variability in the decision. Whichever line comes out on top most is the decision that is most preferred based on all the information entered in. This summary sheet also calculates the Number of Wins for each alternative and graphs them in a bar graph. These summary graphs look something like this:





And that is the program in its entirety. It is a work in progress. The Analytic Hierarchy process can be used in numerous ways, with multiple raters, with sub objectives, and with risk and uncertainty specifically addressed. I fully intend to add this type of functionality into the program in the future, but the program as it stands is a great way for a single user to go through a decision-making process and get visual results of the best decision for them based on their objectives, alternatives, and how important or desirable they feel each of those are. The method is powerful, and having it automated here takes out a lot of the time generally needed to run through it.

## Learning and Conceptual Difficulties Encountered

I learned so much about programming in VBA through this project. A few of the big things I learned concern these concepts:

1. How to use the formula and formulaR1C1 functionality
2. Loops and nested loops
3. Find and replace
4. Ranges
5. Global and public variables
6. Interacting between user forms and sub routines
7. Arrays and dynamic alternatives and objectives
8. Selecting and formatting cells
9. Working with graphs and tables
10. Recording myself doing things
11. Error checking

***How to use the formula and formulaR1C1 functionality:*** These two tools are invaluable to writing formulas. I did not know anything about them when I got started with this project, and I ended up spending hours and hours using them to program all the formulas I needed. Dr. Allen helped me really understand what I was doing with both of them and how they could help me in different ways and I feel that I could use them very easily in future projects.

***Loops and nested loops:*** I had used these in earlier programs I've written, but I had to use them in so many ways and with so many different formulas and other processes, that I really have a better handle on what they can do for me now. There were several times during the process where I spent

an hour or even two hours with my husband helping me go through the logic of a loop to understand why it wasn't doing what I wanted it to do. He has taken a very light VBA class and really helped me with the logic part of the programming. He usually didn't know what to actually type in to make it work, but particularly with the loops he helped me get them to work once I knew what I needed in them.

***Find and replace:*** This was an invaluable tool to use in conjunction with the formula function. I learned that function automatically locks all the cells it enters in, which made it difficult for me to copy some of my formulas in the Alternatives sheet. This was one of the last things I needed to do and I didn't have anyone to help me learn how to use this function, so I actually found out the syntax for it in the Object Browser and VBA help. I took Dr. Allen's advice and browsed through the Object Browser to get my answer and I was very pleased to learn he was right and that I could learn how to do things with this tool.

***Ranges:*** I referenced one of our early class period documents extensively to better understand the multitude of ways to select ranges. I do this probably five different ways throughout my project and I didn't know how to do any of them before I started. This will be extremely helpful in future projects I do.

***Global and public variables/interacting between user forms and sub routines:*** I struggled mightily with taking data back and forth between my user forms and sub routines. I don't know that I learned so much how to do it successfully but more what the limitations were and how I could think outside the one way I thought to get something done in my program through this stressful part of the process. I was so early on in my project and got so frustrated when I could never get the data I needed. It worked out in the end and probably in a much more slick way than my original thought. Where I declared the variables and how I used them made a huge difference, obviously, but I learned so much about it through doing the user forms part of my project.

***Arrays and dynamic alternatives and objectives:*** I felt like I had a pretty good handle on how to use these before starting this project, but I feel more so now. They helped me quite a bit in my process, particularly as I wanted everything to be dynamic. The entire time I felt like I was programming in a really roundabout way in order to have my alternatives and objectives be dynamic. I feel much more able to do this at a future point now that I have done it in so many varied ways in this project.

***Selecting and formatting cells:*** Ranges helped me so much in this project as I already mentioned. Recording myself really helped here as well. I assumed that formatting cells would not be a big deal because it is so easy to do in Excel. It is definitely much harder on the VBA side. I chose to leave the formatting quite basic, and part of it was because of how complex it can get in VBA. It's not hard per se, just takes a lot of programming. So I left the formatting basic so that when I use this program in the future, I can change the colors myself for each person I use it for.

***Working with graphs and tables/Recording myself doing things:*** Recording myself was the only way I could learn to get the tables in I wanted. However, when I wanted to move the tables around, recording myself showed absolutely nothing. This was the biggest thing I could not figure out how to do. In the summary sheet when I put in the two graphs, they show up right on top of each other. I was not able to discover how to have them in a particular spot in the table. I also could not find how to delete those tables out when I clear the sheets in between my projects. I'm sure I will be able to learn how to do these things in the future, but they were right at the end of my project, one of the last things I was working on, and I decided that automating this part just was not worth it at

this time with this project. However, I used the recording function extensively throughout this project and was able to get a lot of things done with it that I would have never figured out on my own. I'm sold on that method of learning how to program, at least for things like formatting that just don't make sense without seeing it.

**Error Checking:** This was the last thing I intended to do with this project, and one where I truly struggled and eventually came up with very little. I wanted to be able to make sure the users entered in a number between 1-9 in an input box and didn't figure out how to do that. I wanted to make sure the users entered a real alternative or objective in the original Buttons sheet, but also wasn't quite sure how to do that. Time was really the factor here. I have no doubt with more time and a little more sleep under my belt, as I continue to tweak this program those will be some of the things I can add in. For now I am content with the fact that I will be using this program myself many times before I give it to anyone else. As I see bugs and errors come up, I am making a list and will add these things in over time. The internet, recording, and help from those that are better programmers than me at the moment will help me get there.

## **Conclusion**

This has been an incredibly valuable learning opportunity. The project I came away with will help me in my own professional pursuits as I do some freelance consulting on decision analysis. Eventually I would like this to be something I can package and give to others to use. Until it is truly ready to do that, it will be a huge asset for me as I work with others to help make their decisions. This has been a great class and I can see through the process of coding this particular process and the learning I gained from it that the more I program, the better programmer I will be, and the more valuable a skill it will be in my toolkit. Here's to learning!