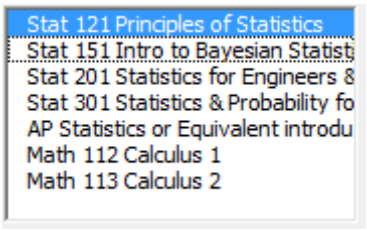
Troy Judd
IS 520 VBA Project
April 14, 2015

Executive Summary

The project is not for a business but is intended to be used by the Statistics Department in counseling students who desire to major in Actuarial Science. Many students want to major in Actuarial Science and need help planning out what classes they should take and when but they cannot use MyMap which is the University provided tool for doing this because they are not yet accepted into the major. In order to be accepted into the major a student must pass the first Actuarial exam on probability theory. This exam is covered by the courses Stat 340 and Stat 370 which are well into the program. Instead students must plan on graduating in one of the other Statistics degree programs until they have passed this first exam. The userforms I have created ask a student what classes they have taken, and what major electives they plan to take and then gives them a list of classes they need to take in the coming semesters in order to give them the direction they need in completing a degree in Actuarial Science.

Implementation Documentation

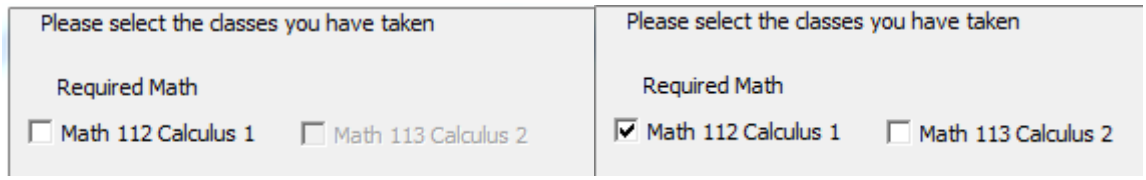
Figure 1 : Listbox method



Originally I thought to have a userform pop open upon opening the workbook that would contain several listboxes containing the required classes and the elective classes within the Actuarial Science major. The user would then highlight the classes in the listboxes accordingly. This method did not seem very cosmetically appealing nor did it seem very functional so I scrapped it deciding instead to use checkboxes.

I created the userform interface with the checkboxes and it worked pretty well but after consulting with Dr. Allen it was decided that a more robust approach was needed to account for future changes in program requirements. Originally, each of the checkboxes with their corresponding classes were hardcoded into the form. This was nice because I was able to use conditional statements within 'click' events that would hide or show a class depending on whether the prerequisite had been checked.

Figure 2 : Example of setting checkbox conditions



These conditions allowed the form to prevent the user from selecting classes that they had not yet taken. The problem with it was that the form could not easily be changed if program requirements

changed. This could happen if classes were added or dropped from the major or if the name of a class changed and so the more robust approach was needed.

To account for changing program requirements, instead of hardcoding the checkboxes, they are dynamically generated according to the classes listed on the Actuarial Program's webpage. I created a macro to query the webpage http://registrar.byu.edu/catalog/2013-2014ucat/departments/Statistics/ActuarialSciMajor.php and pull the classes into an excel spreadsheet. Generating the checkboxes made setting the 'click' events associated with prerequisites as shown in Figure 2 vastly more difficult. The solution to which I was not able to implement.

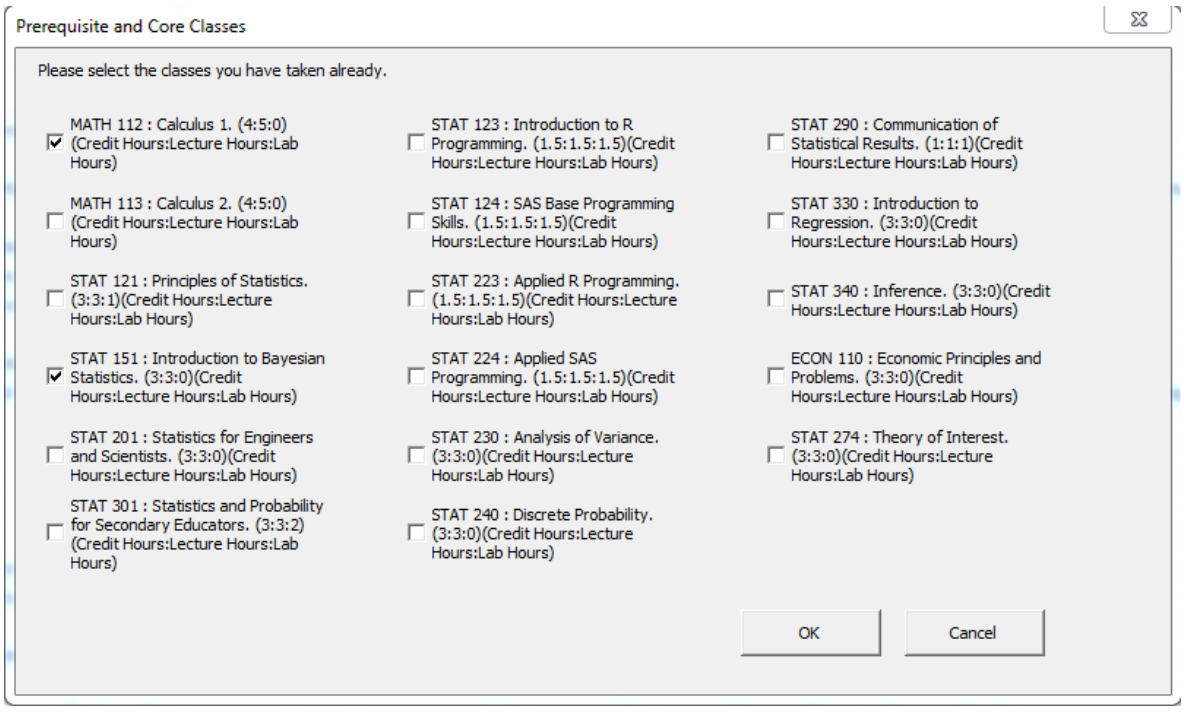
Figure 3 : A Snippet of the data queried

	A	B	C	D
1	1. No more than three hours of credit below C- is allowed in major courses.			
2				
3	2. Students must pass Exam P of the Society of Actuaries (SOA) jointly administered as Exam 1 by the Casualty Actuarial Society (CAS) before declaring an actuarial science major.			
4				
5	3. Complete the following preparation core courses:			
6	MATH 112 : Calculus 1. (4:5:0)(Credit Hours:Lecture Hours:Lab Hours)			
7	MATH 112 : Calculus 1. (4:5:0)(Credit Hours:Lecture Hours:Lab Hours)			
8	OFFERED:	Honors also.		
9	WHEN TAUGHT:	Fall; Winter; Spring; Summer		
10	DESCRIPTION:	Differential and integral calculus: limits; continuity; the derivative and applications; extrema; the definite integral; fundamental theorem of calculus; L'Hopital's rule.		

Figure 3 is a snippet of the data obtained by the internet query. After querying the webpage I chose to generate the checkboxes in three separate userforms.

The first form is simply asking what classes the user has taken. There are certain classes that are required and if the user has not taken them, then a list of those required classes will be generated at the end.

Figure 4 : First Userform



The second form lists elective core classes, of which 12 credit hours must be taken to complete the program requirements. Every class except for the last two which are Research and Academic internship are listed as being three credit hours. For the purpose of this userform they are considered to be three credit hours each and so the user must choose 4 classes to move onto the next form. If the user does not select exactly four classes before selecting OK then a message box pops up telling the user to do so.

Figure 5 : Second userform. Please Select 4 classes!

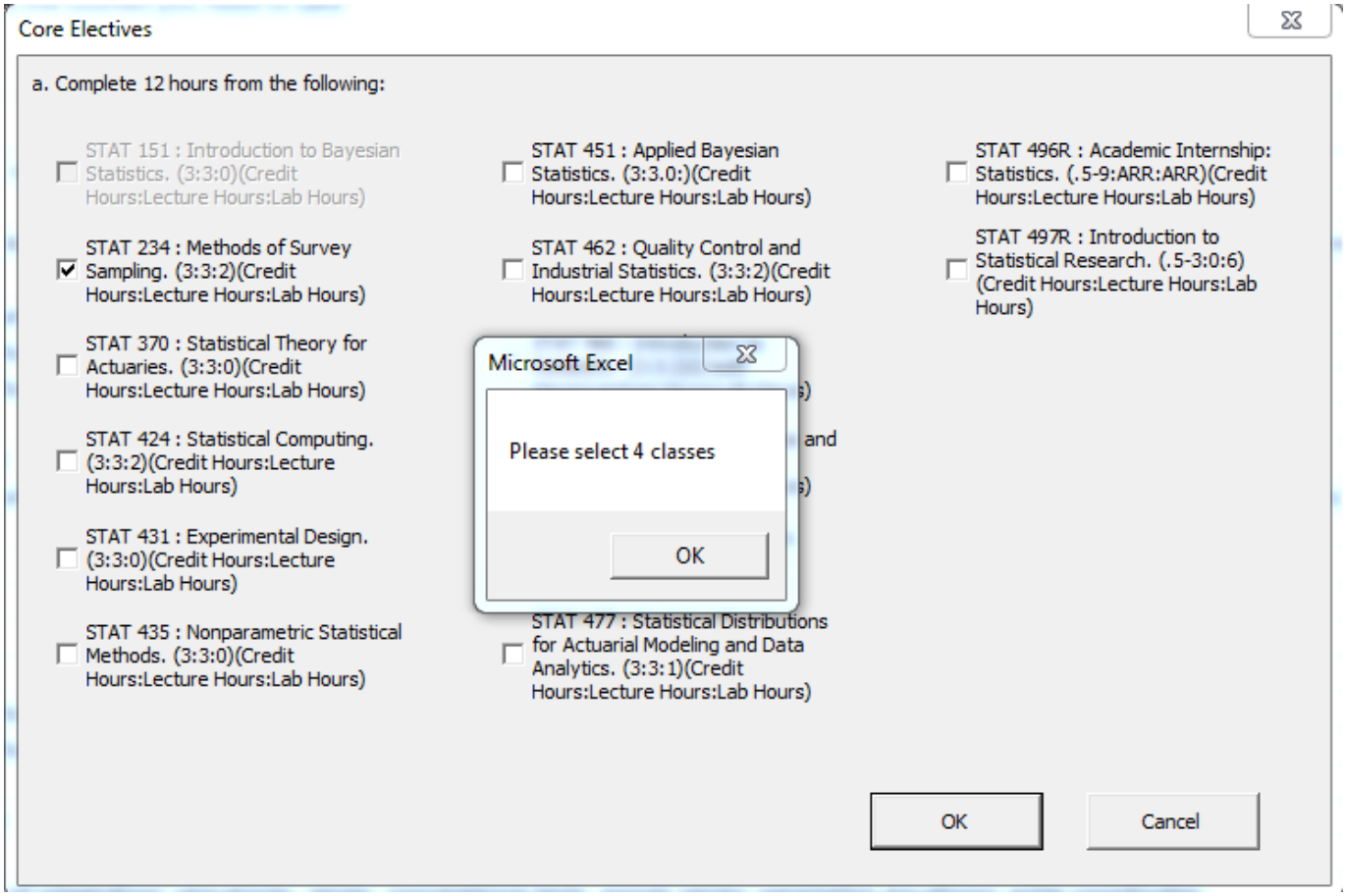


Figure 6 : Second Userform

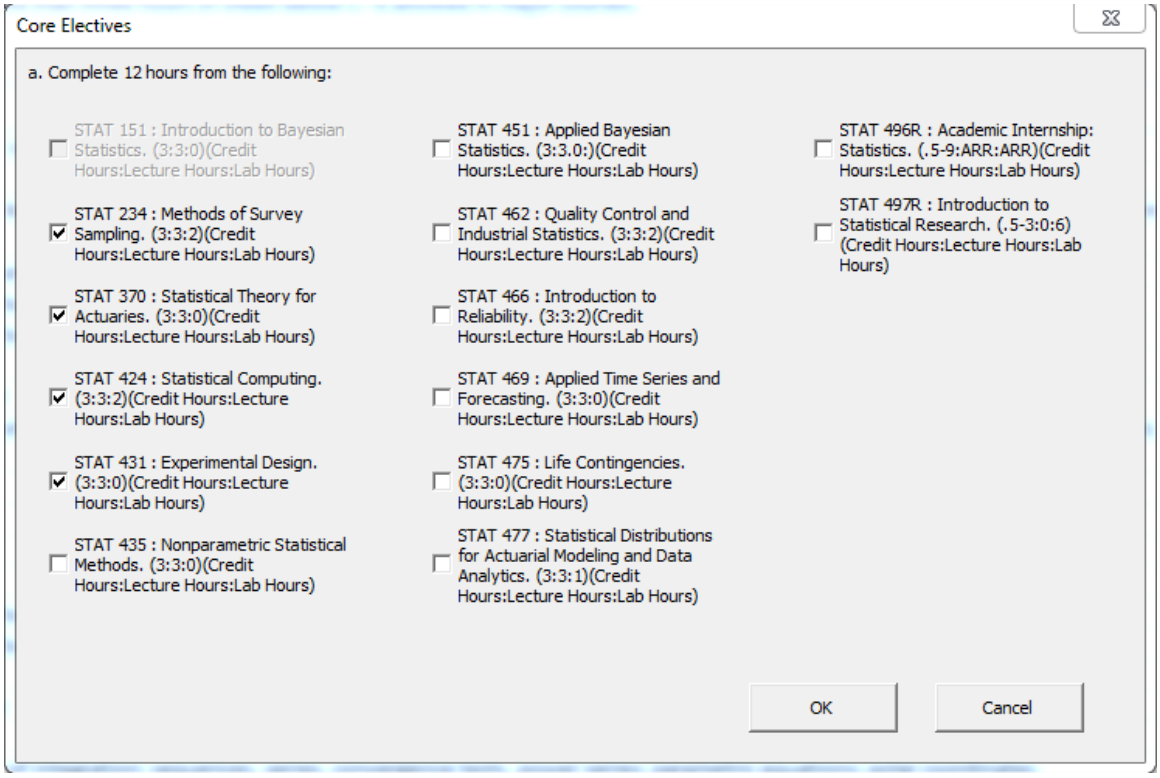
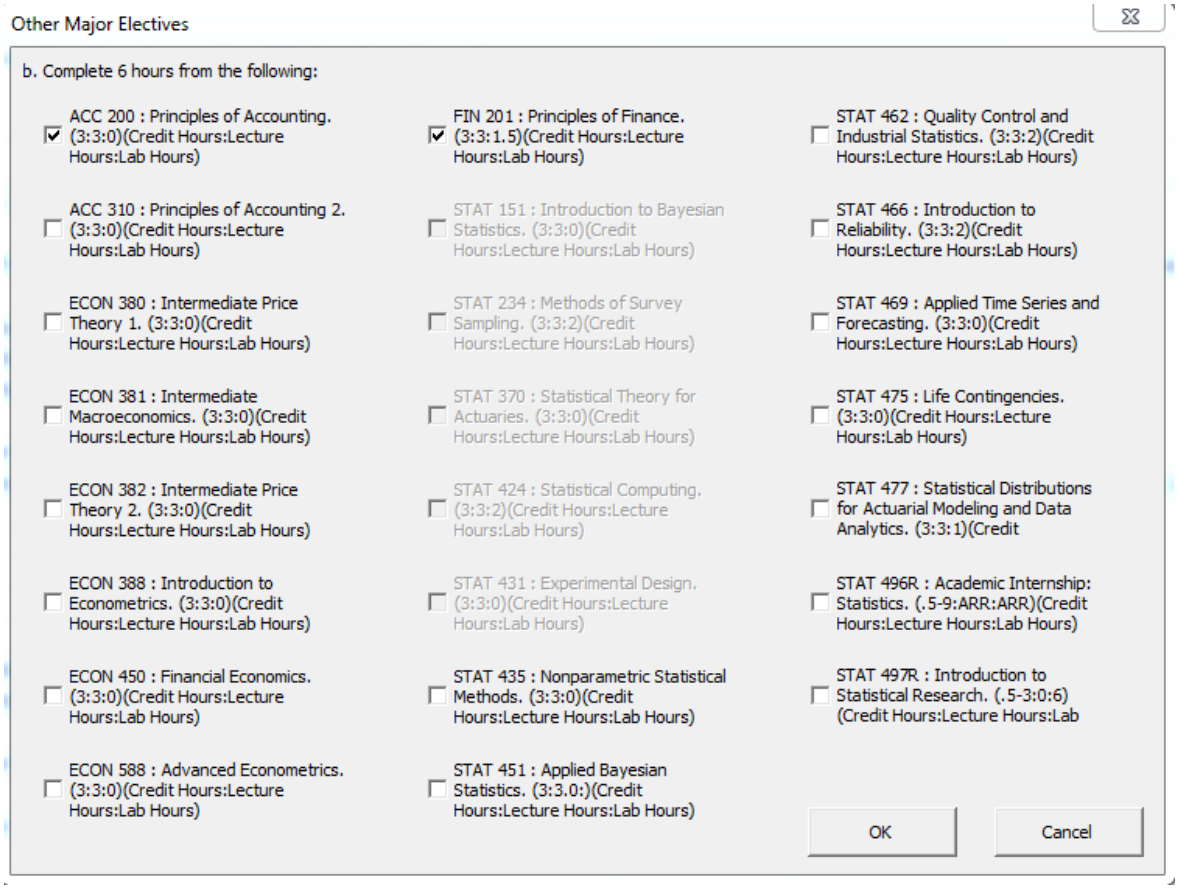


Figure 7 : Third Userform



The third form lists additional classes that may be taken in order to finish the program requirements.

If you look at the second and third forms, on figures 6 and 7 respectively, you will notice that many of the classes on each form are repeated. To prevent a class from being selected twice, the checkboxes for the classes that were selected on the second form are grayed out on the third form and are not selectable. This is done by checking the values of each checkbox on the current userform when the OK button is selected before generating the next userform. If a checkbox was selected on the second userform then it will be grayed out on the third userform.

This was also implemented on the second userform with the option to take STAT 151. In Figure 5 STAT 151 is grayed out because it was selected in the first userform (Figure 4). It is an option in the first userform because it is an introductory statistics course that can take the place of STAT 121.

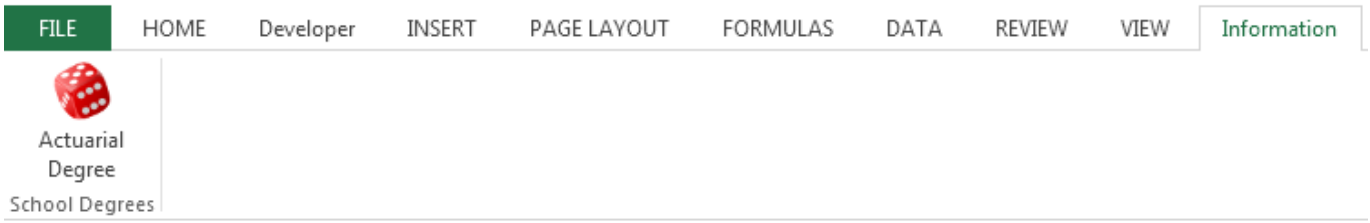
After each userform was executed the classes that the user needs to take are stored in arrays and output to a worksheet called 'ClassesToTake.' The resulting output is shown in Figure 8.

Figure 8 : End Results

	A	B	C	D
1	These are the courses you need to take			These are the courses you have chosen to take
2	MATH 113 : Calculus 2. (4:5:0)(Credit Hours:Lecture Hours:Lab Hours)			STAT 151 : Introduction to Bayesian Statistics. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)
3	STAT 123 : Introduction to R Programming. (1.5:1.5:1.5)(Credit Hours:Lecture Hours:Lab Hours)			STAT 234 : Methods of Survey Sampling. (3:3:2)(Credit Hours:Lecture Hours:Lab Hours)
4	STAT 124 : SAS Base Programming Skills. (1.5:1.5:1.5)(Credit Hours:Lecture Hours:Lab Hours)			STAT 370 : Statistical Theory for Actuaries. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)
5	STAT 223 : Applied R Programming. (1.5:1.5:1.5)(Credit Hours:Lecture Hours:Lab Hours)			STAT 424 : Statistical Computing. (3:3:2)(Credit Hours:Lecture Hours:Lab Hours)
6	STAT 224 : Applied SAS Programming. (1.5:1.5:1.5)(Credit Hours:Lecture Hours:Lab Hours)			STAT 431 : Experimental Design. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)
7	STAT 230 : Analysis of Variance. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			ACC 200 : Principles of Accounting. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)
8	STAT 240 : Discrete Probability. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			FIN 201 : Principles of Finance. (3:3:1.5)(Credit Hours:Lecture Hours:Lab Hours)
9	STAT 290 : Communication of Statistical Results. (1:1:1)(Credit Hours:Lecture Hours:Lab Hours)			
10	STAT 330 : Introduction to Regression. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			
11	STAT 340 : Inference. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			
12	ECON 110 : Economic Principles and Problems. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			
13	STAT 274 : Theory of Interest. (3:3:0)(Credit Hours:Lecture Hours:Lab Hours)			
14				

Last, I also incorporated a ribbon tab with a button called Actuarial Degree with the picture of a die on it. This button will execute the project I created.

Figure 9 : Ribbon button



Discussion of learning and conceptual difficulties encountered

I learned more about worksheets, checkboxes, and userform properties, case statements, for each and for loops, do while loops, variable scope, and strings (Instr, mid, and split functions).

Most of my project dealt with userform properties and checkboxes. The web query was not too difficult but creating checkboxes based on values in the queried data and then referencing those checkboxes was very difficult.

The biggest problem I faced was dynamically associating classes with their prerequisites. My initial approach of hard coding the data would have worked but we decided that a more robust approach would be better. I worked with Dr. Allen a little bit on finding a way to do this. He helped me find the beginning part of a solution which I was not able to implement due to lack of time. Incorporating the prerequisites listed for each course is essential in order to properly generate a path to graduation for the user. Instead my workbook simply tells the user what classes he/she has left to take in order to complete the Actuarial degree requirements.

Assistance received

The only assistance I received was in creating the ribbon tab and button to execute the program and the assistance I received from Dr. Allen concerning the way to incorporate class prerequisites into my workbook.