

Summary: TaxHawk is an online tax filing service here in Provo and this semester they ordered an On-Campus Internship to perform statistical analyses to determine what makes a user of their service either complete their taxes or abandon the process before completing. One of the things they wanted us to examine is if any of the order pattern in which users hit the pages had any statistical significance. This is the problem I attempted to answer. They gave us data from users and the order they hit each page and I created a Visual Basics Sub Procedure that compares each user with all other users in the data set and finds any common patterns in the order they hit the pages. The Sub Procedure asks for a csv file containing user data, then finds the patterns among the users included in that sample, and finally returns a csv file including a list of all users and the found patterns marking the users that followed the patterns.

Implementation: For my solution I first created a sub procedure to be included within my main sub procedure that requests the user to select a csv file with data to be examined for patterns, if they do not select one the current data in the work book should be used. Then I created two easily editable variables to dictate the minimum and maximum size of these patterns that should be found. I also simply analyzed the data and used the findings and the size parameters for the patterns to create two additional sheets within the workbook to include the results to be visible by the user: one for the patterns and one for the users that followed each pattern. After this I coded the most complex part of my code filling these two sheets with accurate information, which required multiple nested loops and if statements. Finally after all this was done, I saved the final sheet of the workbook including the users and which patterns they follow as a csv that can then be used to perform statistical analyses using other software. I added a button to the ribbon to execute the code.

Learning and Difficulties: I solved this problem partially twice before solving it efficiently and completely on my third try. The first try took about 30 minutes to run the code and the second try still took about 20 minutes to run, now each time finishes running in under 10 minutes and most data sets finish running in only two minutes. Through that repeated process I learned about writing efficient code, specifically about having my loops only include things that were necessary to be repeated and leaving everything else out to only be run once. I also learned that there is more than one way to solve almost any problem and sometimes it is worth it to explore those different ways to find the best or fastest one. One of the difficulties I ran into was larger data sets needing variables that were set as integers to be changes to be long variables. Another difficulty for me was in understanding what makes an infinite loop and what does not, I feel I learned the value of exit for and exit do. Perhaps the hardest two things for me to get to run correctly were first preventing my code from recording duplicate patterns and second after finding the patterns in the data recording which users had followed which patterns. Luckily as I solved the first problem I realized it was also the solution to my second one and added lines of code to mark the user that a duplicate pattern was found in the column of the pattern being duplicated.

Assistance: I did not receive any major assistance on this project with the exceptions of the occasional google search and excel macro record button.