FCD Students Attitudes and Behaviors Survey Report Automation

Katherine Hansen

Winona State University

DSCI 495: Communication of Capstone Outcomes

Abstract

Freedom from Chemical Dependency Prevention Works (FCD) administers the Students Attitudes and Behavior Survey (SABS) to measure student attitudes and behaviors regarding alcohol and other drugs as well as their perceptions of schoolmates' attitudes and behaviors. The survey results are then communicated within a 100-page report. The goal of this project was to automate the report. Using Microsoft Visual Basic for Applications (VBA), ways to automate in-text inputs, tables, and charts from Excel to Word were determined. While the whole report was not completed, the completed portion runs much faster than the previous method of manual input.

Introduction

Recurring reports are frequent for many organizations, though the reports are mostly the same, only changing numbers for instance, they still take valuable time to create each report. Once set up, automation greatly simplifies the process while saving an abundance of time. Freedom from Chemical Dependency Prevention Works (FCD), a global non-profit substance abuse prevention organization and part of the Hazelden Betty Ford Foundation (HBFF), administers the Students Attitudes and Behavior Survey (SABS) which measures students' attitudes and behaviors regarding alcohol and other drugs as well as their perceptions of their schoolmates' attitudes and behaviors (Surveys and Assessments, 2016). The school surveyed receives a 100-page report with insights and analysis of students' drug use and attitudes within their own school as well as how they compare to nation-wide norms. The report format remains largely unchanged between reports with only slight variations dependent on the grade range the

survey was administered to, for example. Otherwise, only the data in tables, charts, and text statements change based on survey results, as well as information like the school's name, dates, and other text.

The goal was to automate as much of this report as possible to save time creating future reports and limit human error from manual calculation and input over the summer, beginning with a template, sample data to work with, and some other useful information about the report. For successful automation, solving how to automate text information (e.g., insert the school's name), calculating and inputting data into statements (e.g., #% of students have never had a whole drink of alcohol), calculating and inputting data into tables, and creating charts based on table information was essential. Given the short schedule of this project, complete automation was not expected, but rather enough to set up the HBFF Butler Center for Research (BCR) department with the knowledge to continue and modify the work provided was needed. The overall success of the automation is based on how much of the report gets automated, how fast the automation executes, and the accuracy of the resulting report.

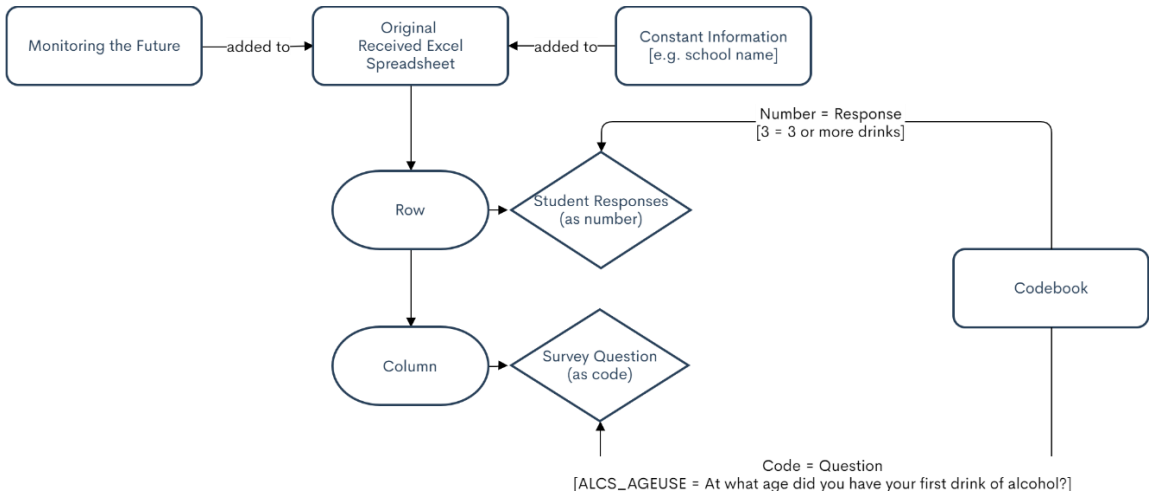
Methods

Previously, the SABS report has been created using a template and then manually calculating and inputting information for the entire 100-pages; the total process spanned, on average, about two months for each report. To instead automate the report, Microsoft's Visual Basic for Applications (VBA) was used because of its easy communication between Excel and Word applications (Microsoft, 2013). VBA is widely popular for automation purposes and granted the data is structured correctly, VBA can automate finding and replacing text, editing tables, and changing charts which saves much time as compared to manual input.

Data

The cleaning and preparation of the survey data is completed before working on the report and comes in the form of a Microsoft Excel spreadsheet. Each row represents a single student's complete survey responses. The columns are the survey questions, each represented by a code. So, a single cell is a student's response to a question but coded as a number. There is a separate codebook documenting what question codes and response numbers correspond to, and so working with the data entails looking up values in the codebook to understand. Also used is data from Monitoring the Future (MTF), a nationwide survey of student drug use conducted by the University of Michigan, which comes from their reports whose data is transferred to an Excel sheet (Johnston, et al., 2020). Finally, another Excel sheet is added to hold data not found within the SABS survey or MTF data such as the school's name, the survey type (paper or electronic), etc. The data on each sheet is in the form of a table. *Figure 1* presents the data collection as a flow chart. The SABS report is in the form of a Microsoft Word document.

Figure 1. Data Collection Flowchart.



The original Excel book consists of rows that contain student responses and columns that contain the survey questions. Both rows and columns are coded and can be decoded using the codebook. The MTF data and other constant information are added to the Excel book as separate sheets.

Find and Replace

Much of the report automation consists of filling in blanks where either text or numbers should go. To automate instances where plain text (e.g., the school's name, first date of contact with FCD, etc.) needs to be placed, VBA has the find and replace objects. Place holders, denoted by the "< >" symbols, mark in the Word template where the VBA macro will input information. *Figure 2* shows an example of inserting the school's name.

Figure 2. Example of Find and Replace Objects in VBA.

```
wdApp.Application.Selection.Find.Execute FindText:="<school>", _
ReplaceWith:=Range("Info[School]"), Replace:=wdReplaceAll
```

In the Word document, all instances of the text "<school>" is found and replaced with the 1st row of the Excel table "Info" and column "School".

For calculations, VBA allows for declaring variables and using normal Excel functions. Declared variables can be used as the replacement object, as shown in *Figure 3*. This is how calculations are automated in VBA.

Figure 3. Example of Find and Replace Objects in VBA Using a Variable.

```
MinGrade = Application.WorksheetFunction.Min(Range("dt[GRADE]"))

wdApp.Application.Selection.Find.Execute FindText:="<minGrade>", _
ReplaceWith:=MinGrade, Replace:=wdReplaceAll
```

The variable *MinGrade* is set to the minimum of column "GRADE" of table "dt". *MinGrade* is then used to replace "<minGrade>" within the Word template.

Tables

The tables in the report are formatted similarly, and the automation solution is to put blank tables with the header and rows labelled like the example shown in *Figure 4* that can be filled in by the macro.

Figure 4. Blank Table in Word Template.

	MTF 8th	<school> 8th	MTF 10th	<school> 10th	MTF 12th	<school> 12th
Total						
Male						
Female						

An unfilled table in the Word template for comparing MTF data and the school's data by sex that is formatted correctly including column names as well as row labels. The blank cells will later be filled in via VBA.

Bookmarking each table makes for easier referencing in VBA. So, the table is found via its bookmark then each cell of the table is filled in one-by-one with the wanted value. Filling a single cell is shown in *Figure 5*. The program cycles through all the table cells in a similar fashion.

Figure 5. Example of Changing a Single Table Cell.

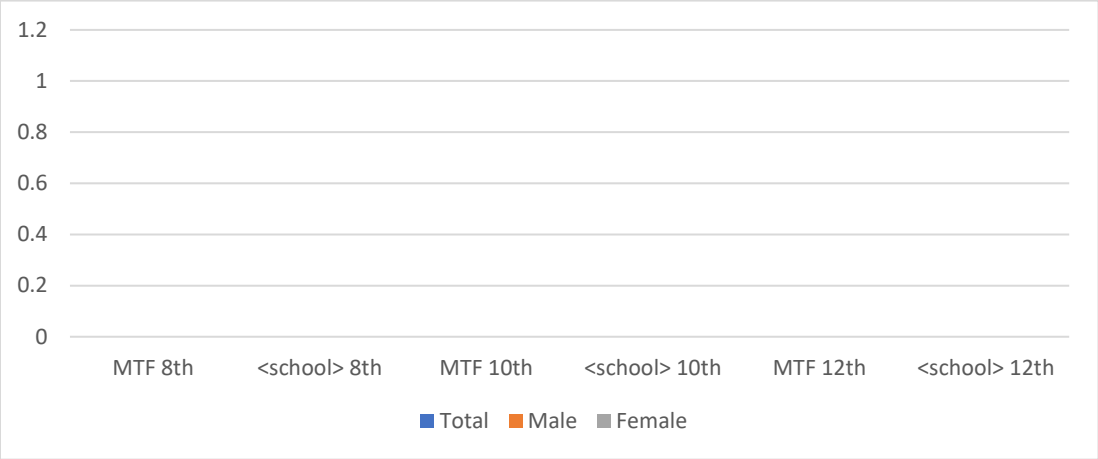
```
wDoc.Bookmarks("AlcYearTable").Range.Tables(1).Cell(2, 2) _
.Range = ThisWorkbook.Worksheets("MTF 2019").ListObjects("MTF") _
.DataBodyRange.Cells(1, ThisWorkbook.Worksheets("MTF 2019") _
.ListObjects("MTF").ListColumns("AlcYear8").Index)
```

Cell (2, 2) of the Word table "AlcYearTable is filled with a value from the table "MTF" on the Excel sheet "MTF 2019". The value at row 1 of the column "AlcYear8" is inputted into the Word table cell.

Charts

Automating charts is very similar to tables. A blank chart with the correct number of bars, labels, and aesthetics in the Word template can be edited using VBA. An example is shown in *Figure 6*.

Figure 6. Example Blank Chart Prepared for Editing.



An unfilled chart in the Word template for comparing MTF data and the school's data that is formatted correctly including column names as well as legend labels. The blank columns will later be filled in via VBA.

Charts are referenced by index: the third chart in the document is Inline Shape 3. The charts present the exact same data as the tables, so the macro references the data in Excel in a similar fashion. Word charts are based upon Excel charts, meaning that the data used in a Word chart is held in an Excel spreadsheet (saved as part of the Word document). The general idea of updating Word charts is to open the Excel spreadsheet tied to the chart, update the spreadsheet data, and then close it. *Figure 7* shows a VBA example.

Figure 7. Example of Changing a Column of a Chart.

```
(a)  Set shp = wDoc.InlineShapes(3)
     Set chrt = shp.Chart

     chrt.ChartData.Activate
(b)  Set wb = chrt.ChartData.Workbook
     Set SourceSheet = wb.ActiveSheet

     SourceSheet.Range("b2").Value2 = wbA.Sheets("MTF 2019") _
(c)  .ListObjects("MTF").DataBodyRange.Cells(1, wbA.Sheets("MTF 2019") _
     .ListObjects("MTF").ListColumns("AlcYear8").Index)
```

(a) The program first finds the chart to edit, in this case the third chart in the Word template. (b) Then it opens the spreadsheet to edit. (c) From here, the program runs similarly as it does to fill tables. To input the same MTF value (row 1 of column "AlcYear8") from the Excel sheet "MTF 2019" with the table "MTF" into cell B2 of the Word chart spreadsheet.

Results

As of August 9, 2021, the first 22 pages out of the 98 total pages are completely automated. Running the VBA macro takes about 5 seconds; if this rate is assumed constant, it can be expected that the fully automated report should be generated in under a minute. This successfully saves weeks of time compared to manual input. The find and replace objects allow for automating in-text changes for information like the school's name or specific data points. Tables can be automated by creating an unfilled table in the Word document which then can be filled-in cell by cell via VBA. Lastly, Word charts can be automated similarly to tables by creating a blank chart and then editing the data in the source sheet. While the report automation is not completed, all cases of automating input are solved and only need to be implemented throughout the rest of the report. As of now, some calculations may be incorrect and will be reviewed by someone more familiar with the report to determine the accuracy of variables.

Conclusion

The SABS report is a very large and comprehensive report to automate. The total project was unable to be completed but is set up as a good foundation for continuation. What automation is in place now runs very quickly and succeeds in saving time otherwise used to manually input information. Somewhat unexpected was the depth of knowledge about the report that would be required, so while an outline was created for the VBA program, knowing what information to plug into formulas, or how to even obtain the information in the first place remains to be reviewed. This gap in knowledge attributes to why little progress of the report was completed; the automation process would have benefited greatly if an individual experienced with the SABS report contributed more information. A lot of time was spent trying to figure out what calculation to perform to get the information that was believed to be what the report was trying to communicate. For now, the provided VBA automation methods were passed to the BCR to finish the report automation along with documentation of the code.

As with most other programming projects, much time was spent also trying to shorten the code: make it more readable and more efficient. A balance was hard to strike between better coding and progress. In the future, using more user defined functions to reduce clutter and more easily modify the code would improve the automation. Reducing redundant code, especially pertaining to tables and charts would also help. There are some instances in place already where an original 16 lines of practically identical code of inserting values was reduced to 6 lines using loops, but this could be reduced even further. The organization of code also has room for improvement. It is not always clear what in the template the code is manipulating and navigating the ever-increasing code can be difficult. Finally, the report changes based on how many school grades are being reported on and it would have been great to have time to put some conditionals

into the automation like to delete the 8th grade columns in tables if 8th graders were not surveyed instead of being left blank.

As automation progressed, it was clear that the structure and contents of the report itself could use more consideration. FCD uses the fact that the report is a 100-pages as one of their selling points, but currently it is very bloated and lacks substantial analysis. The report has a lot of repetition, for example, there is a key points section at the beginning of the report that is seven pages long where the points are a complete copy from within the report. Another example is that the charts show the same information as the tables, a total redundancy. Not only is the report repetitive, but also the overall structure of the report weak. The report is on actual student behaviors compared to their perceptions of peer attitudes towards drug use, but rarely in the report are the behaviors and perceptions data side by side, rather mostly separated by 50 pages. This makes it very hard to connect the data and thus the report loses its focus. Lastly, the way much of the information is communicated is in long and wordy sentences that easily get confusing to decipher. These concerns, and others, were expressed and if acted on, the VBA program would need to change with it. Overall, there is a lot to consider when automating reports, and it helps to know the in and out of the report while working with it.

References

Johnston, L. D., Miech, R. A., O'Malley, P. M., Bachman, J. G., Schulenberg, J. E., & Patrick, M. E. (2020). Demographic subgroup trends among adolescents in the use of various licit and illicit drugs 1975-2019. *Monitoring the Future Occasional Paper No. 94*, 842.

Microsoft. (2013). Visual Basic for Applications (Version 7.1).

Surveys and Assessments. (2016). Retrieved from FCD: https://fcd.org/what-we-do/surveys-and-assessments/