Analytics

Nathan Allred

MBA 614, VBA Project

12/09/2010

Executive Summary

As an MBA student I wanted to learn VBA in order to save time. After graduation, I'm sure that my work will include routine, repetitive tasks. By learning how to automate some of these tasks I will add value at my work and make myself more marketable. One such task that I currently perform on a regular basis is to download and analyze data from a personal budgeting website called mint.com. Mint does a great job of downloading checking and credit card account transaction data and bringing it into one location. However, its data analysis tools aren't complex or sophisticated enough for the type of analysis I want to do. Once I download the data from mint, I bring my budget numbers and transactions data together and put them into a pivot table. I then create several custom pivot charts. Depending on how many spending categories I want to create, this task can take more than an hour. I decided that this would be a perfect task to automate with VBA. The completed program accomplishes this task with the following steps:

- First, the program logs on to the mint website.
- Once logged on, it imports budget data and downloads all transaction data.
- This data is imported to my budget worksheet and put into a pivot table.
- Five separate buttons execute macros that create custom charts

These charts allow me to analyze my spending behavior compared to other categories or actual budgeted amounts. This program reduces my routine task from an hour to around two minutes. I'm excited to use this program every two weeks as my wife and I set our budgets and analyze our spending behavior. I'm also confident that I can use a modified version of the program as I start a new job in finance with American Express in New York City this next summer.

Introduction

One of the reasons I decided to quit my job and return to school to get my MBA was that I wanted to gain practical skills that would make me more valuable to potential employers. I work in finance within the banking industry and much of the work that I do is in Microsoft Excel. I first learned about the power of VBA when my past employer was able to automate the majority of the tasks that our department performed at month end. This cut the closing process from two weeks down to about two days. For my final project I wanted to write a macro that saved me time. I also wanted to write a macro that I could potentially use in the future to save time and increase efficiency at work.

As soon as my wife and I decided that I would quit my job and return to school, we knew that things were going to be tight financially. I would have no income and my wife would only be able to work part-time, so our primary source of income would be savings and student loans. We decided that we would have to become very good at budgeting. I have a friend who recommended a personal budget website called Mint. He told us that it was free and that it did a pretty good job tracking expenditures and managing your budget. We decided to sign up and give it a try.

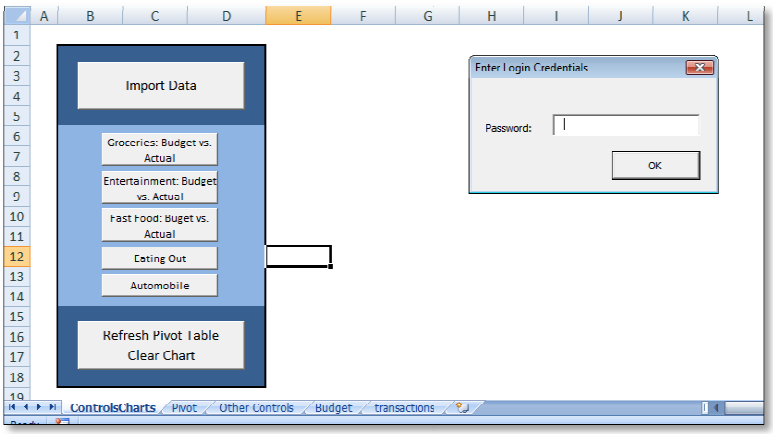
I loved the fact that Mint was able to aggregate transactions from all our bank accounts and categorize them. The budgets were really easy to set and change and Mint even notified us when we overspent in a category. However, one problem I always had with the website was that I wasn't able to analyze with very much detail our spending history and behavior. I wanted to do this for two reasons: First, I wanted to be able to see what we historically spend in each category in order to set achievable spending goals. There's nothing worse than overspending month after month in the same categories. Also, I wanted to have a dynamic and understandable way to look at past spending trends and see where we needed to cut back. Mint does have a trends tab that gives you the ability to look at spending in each category over time. However, I wanted to be able to have pivot table functionality in order to compare historical spending to actual budgeted amounts. I also wanted to be able to see several different spending categories in the same graph. For example, I wanted to be able to graph all automobile expenses over the past year including insurance, gas and maintenance.

Because mint allows you to download all past transactions in an excel file, I started analyzing the information on my own. I would download all my transactions and create a pivot table. In order to graph a particular spending category against the actual budgeted amount, I had to download the transaction file and then manually enter actual budgeted amounts into my data file for each month I wanted to graph. With that pivot table, I would choose the dates and categories until I had a suitable graph. The entire process was taking over an hour twice a month when my wife and I set our budget.

After learning about VBA in MBA 614, I was pretty sure this was a task I could automate. After graduation I will be working in corporate finance at American Express. One of my monthly or bi-monthly tasks will be to download and analyze economic and competitor data. I felt like this program would work perfectly for this kind of task with some small modifications.

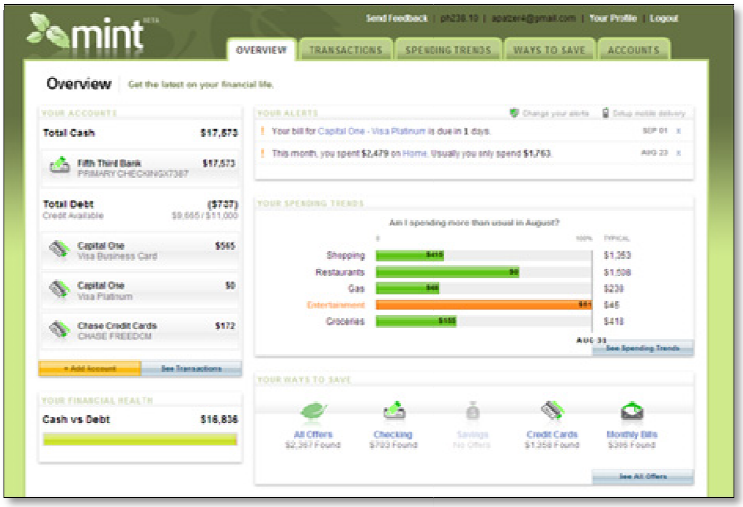
Project Description

The first step of my project was to log in to the Mint website. I used the functionality of the class module, "agent" to perform this task. When you click the import data button, a very simple user form pops up asking for your password. When the password is entered, the macro logs you into the site. The first thing it does once logged in is to import the overview page using agent1.importPage



"sheet". I wanted this page imported to my workbook because it has my actual budget data by category. I would later add this to my transactions data for analysis. This data is imported to my workbook as a new sheet called "budget". Before it's actually imported though, the old budget sheet is deleted.

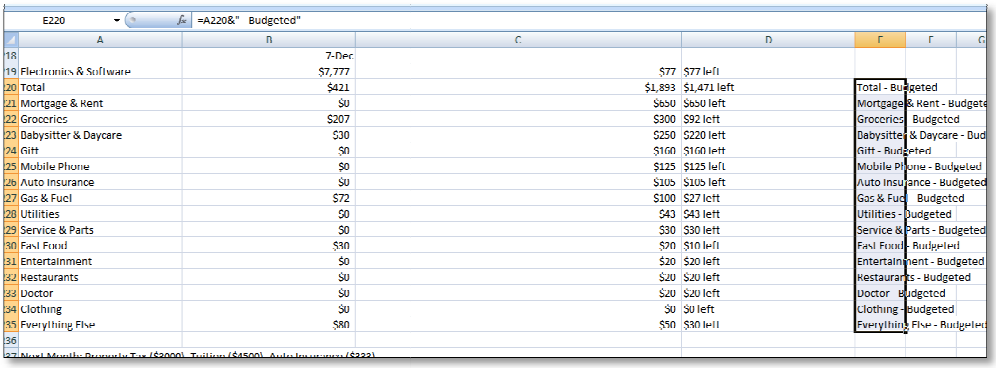
After this data is imported to my workbook, agent1.openpage brings me to the page where I can see the link to download all transactions. I then tell VBA to click on that link. When I ran this code for the first time, an information bar popped up and wouldn't allow the file to download. With a little research, I found out that this was because, by default, Internet Explorer doesn't allow automatic prompting for downloads. I was able to fix the problem by enabling automatic prompting in tools> internet options> security. I even wrote a simple procedure using the send keys function (I know this is unreliable, but I couldn't find another way to change the settings in Explorer with VBA) to enable prompting very quickly. This can be found on the sheet titled, "other controls" and only needs to be used when the program is being run on your computer for the first time. Alternatively, you can manually enable automatic prompting prior to running the macro.

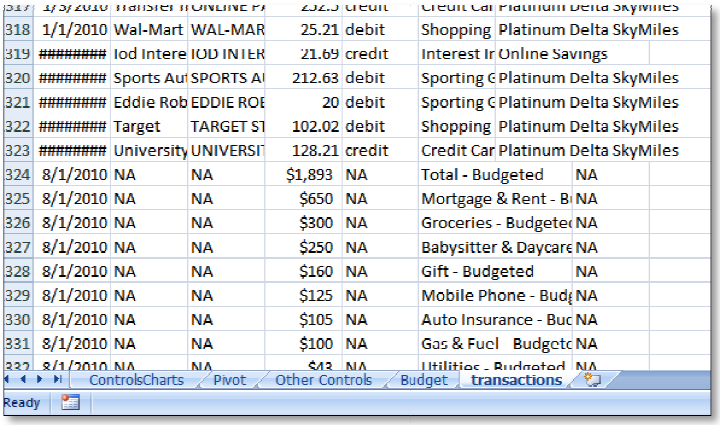


Once automatic prompting is enabled and the transactions file is downloaded, I needed to import the data to my workbook. I decided the best way to do this was to move the whole sheet using sheets(sheetname).move rather than copy and paste the data. However, I ran into a problem. Every time I downloaded this file from the internet, it had a different name. I wasn't sure how to activate that workbook when it had a different name each time. I hoped that since it would have just been downloaded it would already be the active workbook. However, when I tried to run all

the code together, it didn't always recognize it as the active workbook. After much searching and not much finding, I was got the answer from my professor. The most recently opened workbook could be referenced with workbooks.count. With this, I was able to first delete the old transactions sheet, then select the recently downloaded workbook and import the data onto a new sheet called "transactions" in my project workbook.

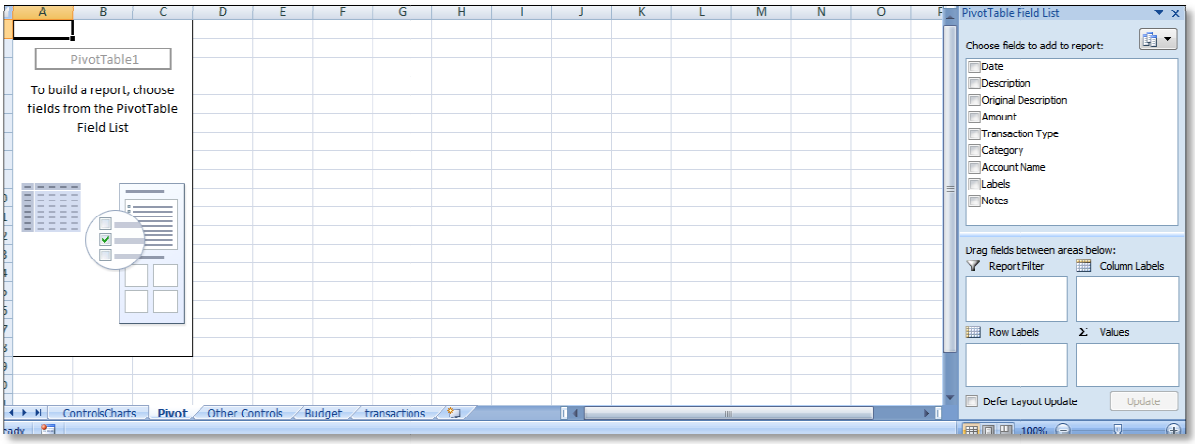
With all relevant data in my workbook, I was ready to build my table. First I needed to find my current budget data. To do this I told VBA to go to the budget worksheet and search for the word "total". This word is found in this data only once and is located just above my budget data. From that cell, I use relative references to select the cell that contains my first budgeted category. I then move to the right of that and enter a formula that will add the word " - budgeted" to the name of the category (see image below).





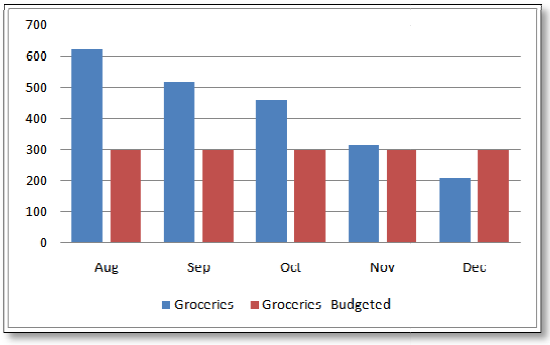
I then tell VBA to copy and paste the category names along with their corresponding budget amounts to my data file as shown in the image to the left. In order to complete the necessary data for my table, I tell VBA to enter dates for each necessary month in the date field and the value NA in all other fields.

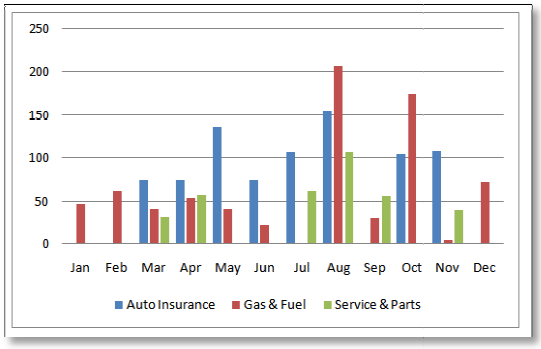
With my date table complete I was ready to actually build the pivot table. This was relatively easy to do using ActiveWorkbook.PivotCaches.Create. I did have to make sure I deleted the old pivot table before building the new one.

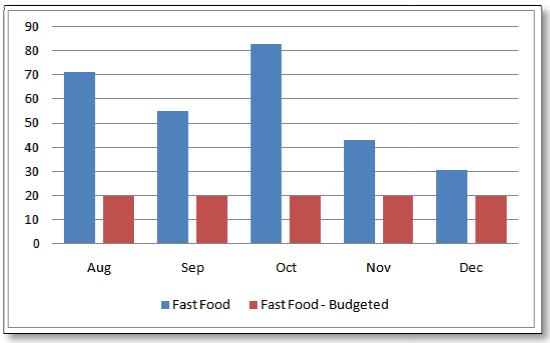


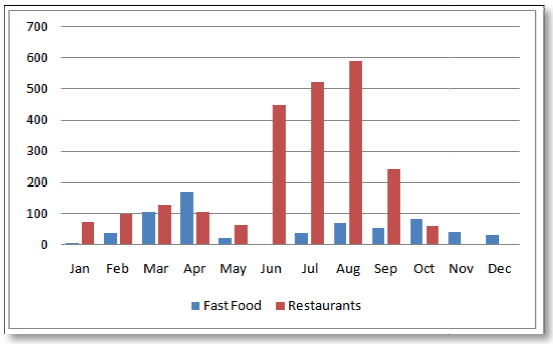
With my pivot table in place I was then ready to build my charts. I wanted to have several charts. Some of the spending categories that we were watching the closest included groceries, eating out, fast food, auto expenditures and entertainment. We felt like we had the greatest opportunity to cut spending in these areas. Therefore, these were the graphs that I wanted to build first. I needed to come up with the code for a procedure that would select all the desired dates and categories in my pivot table and create a chart. I found that the easiest way to do this (since I had already spent close to 40 hours on this project, I wanted to find the quickest way to do this) was to record myself building the chart and then modifying the code to do exactly what I wanted.

When I recorded myself, VBA created a "with" statement and then went through each category that I un-clicked and set that category to false. The part of the procedure that I had to write was to delete the old chart and reset the pivot table first and then move the newly created chart to my controlsCharts sheet. I repeated a similar procedure for each of five charts for this project. The following images are the charts the various macros generate. As you can see, they are just the standard pivot chart templates that excel generates.









Next Steps

Generally when I build charts and graphs, I format them a certain way. I do a gradient fill and lighten the gridlines and size the graphs so the data is easy to see. The graphs that are generated by this program are very generic and aren't anything special. When I tried to record myself formatting the charts, the code wouldn't run properly. My next step will be to learn to write code that will format these charts the way I want. I'll change the colors, sizing, fill, overlap, and several other variables. I believe this will make this tool much more effective and professional looking. This way, when I modify this project for my future job, I will be able to quickly import these graphs to power point, as that is how presentations are made at American Express. Overall, I'm very pleased with the outcome of the project and what I've learned along the way. I was able to build something that automates a task. What used to take me about an hour now takes about 2 minutes. I'm excited to use this and other VBA tools as I re-enter the workforce.