|  |  |
| --- | --- |
|  | **2010** |
|  | MBA 614 – Spreadsheet Automation and ModelingBritt Chapman |

|  |
| --- |
| **[Comparable Company Automation Project]** |
| A tool to help investment professionals and students quickly download and organize key financial data for comparison and valuation purposes.  |

**EXECUTIVE SUMMARY**

*THE PROBLEM*

Most professionals in the investments industry (investment banking, private equity, investment management) and the valuation business (accountants, actuaries, fair value specialists) use a program called Capital IQ in order to quickly pull lots of information on public companies for comparison and valuation purposes. Capital IQ has a special excel spreadsheet that can be downloaded and used by these professionals. The spreadsheet interfaces with the Capital IQ website and quickly pulls financial data on any public company that you type into the spreadsheet. Please click on the link below to become familiar with the Capital I.Q. excel plug in and the importance of having this data automatically generated. Also, open the PDF brochure to see part of an excel worksheet that I am trying to emulate.

<https://www.capitaliq.com/Main3/ourproducts_platform_excel.asp>

There is only one problem. Capital IQ is outrageously expensive for students who often need the same tool. Luckily, most of the core data is available on Yahoo.com or other websites. It just needs to be pulled, sorted, formatted, and manipulated.

*THE SOLUTION*

I used VBA to set up a spreadsheet that allows the user to enter ticker symbols of public companies for comparison purposes. I originally set out to have the spreadsheet automated to the point that the user could put in the time period that is desired, the base for the beta analysis (i.e. S&P 500, Russell 3000, or Wilshire 5000), the time interval for the beta calculation, the currency denomination, and other variables. I also wanted the spreadsheet to pull and organize every piece of data under the sun including the companies’ stock volatility, historical working capital, historical capital expenditures, etc. I soon learned that there is a reason that Capital I.Q. is so expensive. In order to completely re-create the Capital I.Q. excel plug in would demand about a year’s worth of work for one person. Therefore, I chose a couple of pieces of the project that I could complete and left the rest for a later date.

The resulting spreadsheet automatically downloads data from Yahoo Finance upon the user manually entering the ticker symbols of the companies that are comparable to the users target company. The spreadsheet then organizes this data for each comparable company and provides the user with a quick way to compare the following data to the target company:

* Historical Beta
* Leverage ratios to automatically un-lever and re-lever beta
* Current P/E ratios, and Enterprise Value ratios
* The last three years of income statements
* Current Stock Price and Market Capitalization
* Current Cash on Hand
* Company description and applicable markets/industry

The best part about the spreadsheet is that from here, it will not take me much longer to have the spreadsheet automatically calculate a WACC for my target company. This will greatly benefit me next year in my finance classes. I can just input the assumptions, type in several comparable companies and the date of my analysis, and poof, my WACC will be provided and formatted.

**IMPLEMENTATION**

*USER FORM AND MAIN PAGE*

I began my project by creating a user form that would allow the user to input various assumptions to receive the correct data. Below is a screen shot of the main tab of my spreadsheet and another screenshot of the user form.



The top portion of the main page deals with the user form and the user options while the bottom portion deals with the input of ticker symbols and the generation of the data which we will discuss below. The most difficult item in this stage of the project was the calendar. I didn’t have the desired calendar installed on my computer and so I had to google user form calendars until I found and downloaded exactly what I wanted. It was refreshing to learn that I could get what I wanted on my own. I have no computer science or programming knowledge at all and doing this part on my own was a big step up in my confidence. It should be noted that the user form opens with whatever options are already chosen when the “Edit Inputs” button is pushed and it updates all of the cells when the inputs are changed and the “OK” button is pushed. It doesn’t change anything when the “Cancel” button is pushed. There are six options that can be changed: Start Date, End Date, Beta Interval, Beta Base, Currency, and US Treasury Data.

Unfortunately, I built this great user form to provide the user with several options in data generation and didn’t use any of the user form options on the rest of the project. I just didn’t get far enough along on the spreadsheet in order to build out the application of any of these options. I still have many more hours of programming before I get to that stage.

*TICKER SYMBOL AND GET DATA*

The next stage of the project involved including a place for the user to input ticker symbols of comparable companies (in finance lingo these companies are called “Comps”). After the ticker symbols are entered, the user may then click the “Get Data” radio button which will kick off a long series of commands.

First, the spreadsheet will automatically insert a worksheet to hold the data of the Comp. The new worksheet will automatically be formatted and key data will be brought to the top for analysis. This required pulling lots of data from several different web pages from yahoo finance. I decided to import the whole web page for each of the different web pages and place all of the data at the bottom of the worksheet beginning at row 10,000 and going in increments of 10,000. For example, the income statement data is imported and placed at row 50,000, the balance sheet at 60,000, and the cash flows statement at 70,000. Below is a screenshot of the resulting new worksheet.



There are a few improvement items that are needed like the EV / EBITDA result being aligned right and the market capitalization being formatted with commas in the proper spaces (Throughout the project I place high emphasis on formatting. Even though this is not the most difficult part of programming or VBA, it is vitally important to investment professionals who are obsessive compulsive about formatting. The formatting enables quick comparison and establishes professionalism). However, I am pleased with the overall result.

The most difficult programming of this stage involved searching for key words that would occur in every download and would provide a base cell from which I could find the data that I needed. At one point, I had to add the short term and long term debt in order to get total debt at the top. If the company did not have any amount in either the short term or long term debt, then yahoo automatically places a dash in the cell. This dash created a #Value! error in my total debt cell even though there might be some debt. I used the below if statement to fix this problem.

' Get Total Debt

 Cells.Find(What:="Short/Current", After:=ActiveCell, LookIn:= \_

 xlFormulas, LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:= \_

 xlNext, MatchCase:=False, SearchFormat:=False).Activate

 ActiveCell.Offset(0, 1).Range("A1").Select

 Application.CutCopyMode = False

 **If Selection = "- " Then**

 **Selection.Clear**

 **End If**

 Selection.Copy

 Range("AA1").Select

 ActiveSheet.Paste

Second, the spreadsheet will then automatically run through all of the ticker symbols pulling the data and formatting the worksheets. This is all done behind the scenes as the main page is the only thing that the user sees. The main page will also be updated with the names of the companies as seen on the first diagram of the main page.

Third, the spreadsheet has a summary tab that is prepared to receive all of the data and format and calculate the data into a form that is conducive for comparison. The summary tab is shown in the below screenshot.

The most difficult portion of this step was ensuring that the summary worksheet was dynamic enough to handle any number of ticker symbols that might be entered by the user. The coding follows the below logic:

1. Insert worksheet and update and format worksheet for the first Comp.
2. Copy and paste the data into the summary worksheet on row number 7.
3. Add applicable formulas for comparison.
4. Insert row above row number seven which will move the first Comp to row 8.
5. Copy and paste the company name onto the main tab.
6. Move to next Comp.
7. Start over at step number one.

Step number 2 above provided a significant challenge due to the difficulty of automatically moving back and forth between tabs. The name of every Comp’s tab will be different, so how could I make one code that would work in commanding excel to go back to the current Comp’s tab to copy another cell? For this dilemma, I created a public variable at the top of the module (variant because you never know what ticker symbols you might get):

**Public x As Variant**

Then make x equal to the ticker symbol in the beginning stages of the code:

**x = Range("A1").Value**

Then name the worksheets the same name as the ticker symbol:

**Sheets(Sheets.Count).Name = Range("A1").Value**

Then have the toggle back and forth between the two tabs just reference the Comp tab as a worksheet with name x.

**' Copy Beta**

 **Range("C7").Select**

 **Application.CutCopyMode = False**

 **Selection.Copy**

 **Sheets("Summary - Do Not Delete").Select**

 **Range("F7").Select**

 **ActiveSheet.Paste**

 **Sheets(x).Select**



\*Note that the EV / EBITDA above is N/A because I pulled the data for a lot of banks for my example.

**LEARNING AND CONCEPTUAL DIFFICULTIES**

Contrary to my prior belief, this project actually taught me more than any other project in my classes this semester. I thought that I would really struggle with the concepts and the coding, but the more that I played around with each step, the better that I got and the quicker that I figured things out. It was slow going at first. As mentioned, I don’t have any background in IT or programming and so this was a huge victory for me. The most important thing that I gained was confidence that I can do now figure this stuff out on my own. Moreover, the most important concept that I learned was that it is crucially important to first decide on the logical steps and sub steps of your project so that you lay out the code in the order that works. The order that the code runs is vitally important.

There were not any bugs or difficulties that I didn’t figure out on my own. I thought that I was going to need to get some help on a few different sections (see the difficult areas mentioned in the write up above) but I just kept changing the different logical steps around until I stumbled upon one that worked and that I knew how to code. The one disappointing result of my project was that I really only finished one small part of my grandiose idea. I am glad that I now have a tool that will help me to quickly pull some Comps and get a median Beta, EV/Revenue, and EV/EBITDA for my analysis, but I would also love to quickly get a WACC, capital structure, working capital %, and capital expenditure %.

Another surprise during the project was how much time it takes to code and correct the formatting. Even if you run a macro for the formatting, you still have to place all of the code in the right order and tweak many parts of the code to generate exactly what is needed. A simple macro will not get the job just right. Even though the formatting took a lot of time and probably does not generate the “wow” factor for my grade, I still insisted on doing it because I wouldn’t use the tool without it.

Thank you for the opportunity to do this project and if you have the burning desire to get some free Enterprise Value / Revenue ratios, then open up the tool and enjoy.