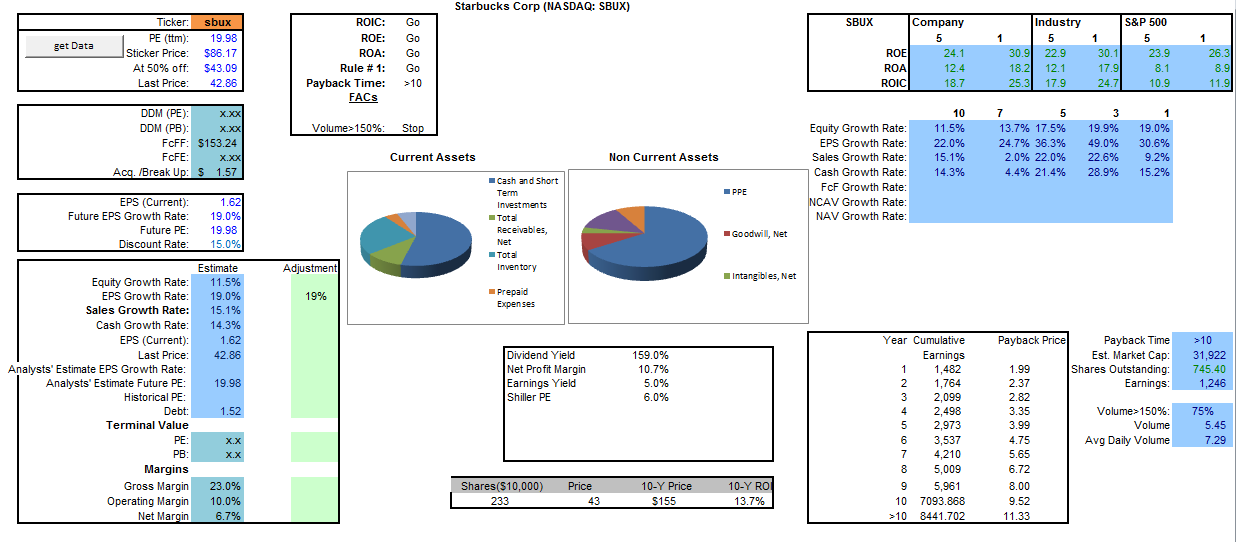
**ISYS 520 – Spread Sheet Automation**

Instructor: Gove Allen

Equity Analysis Tool



C. Roberto Lall

Friday, December 09, 2011

Executive Summary

The Problem:

I want to form opinions about publicly traded equities quickly. I participated in a student run investment fund, Global Financial Advisors. I listen to student presentations on firms operating across the market. We discuss these firms at length. I like many students, balance listening to the presentation with pulling up financial data, charts, and qualitative information. The financial data is spread across several webpages. The basic data is available, but it isn’t formatted to be intuitively meaningful.

I also like to paper-trade options. Often the trade is quick and exploits a technical chart pattern. In this case, as well as the first, it is helpful to have a snapshot of the company that draws a picture of the company fundamentals. For example, if I see positive signal in the market for the name it would be helpful to know if the company was over-levered first. Conversely, if the name is significantly undervalued I would want to know before selling it.

The Solution:

This workbook consolidates the information that I like to have available to form an opinion about a publicly traded equity. The primary purpose of this tool is to articulate meaning with the data. It does this in four ways:

* It only shows essential data on the main tab.
* It organizes tables more efficiently.
* It converts raw data into ratios that convey meaning.
* It consolidates balance sheet data into charts.

The fundamental questions it answers is,

“Should I look at this company further?” & “What should I look at?”

It is not intended to answer the question, “Should I buy this stock?” So, if you use it in that way it isn’t my fault if you have a bad experience.

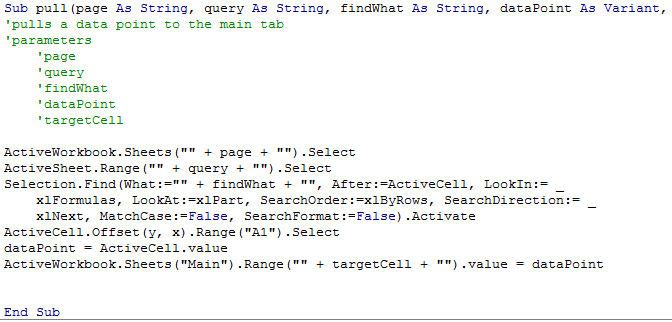
Implementation

This project started in late August of this year. This is the second major iteration. Over the summer I read two books by Phil Town: Rule #1, and PayBack Time. The first iteration only contains data and methods expressed in his books. This iteration has built on that framework and extended it to include more data and other methodologies.

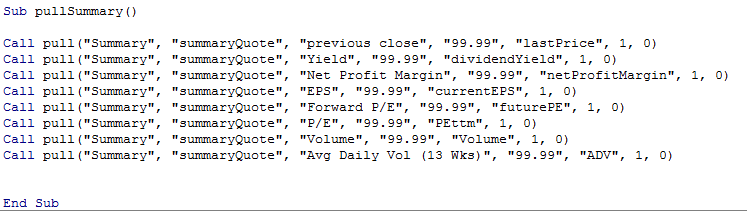
Modular Structure & Populating the Dashboard

The program populates the dashboard with data from queries of MSN Money. The sub-procedures ‘pullSummary()’ & ‘pullFundamentals()’ perform this function. The code is separated into 3 modules. The first module holds all the primary sub-procedures involved in the process of getting and analyzing the data. The second holds a secondary sub-procedure called, Corral the Bulls(). The third module contains sub-procedures that support the primary and secondary procedures.

For example, the pull() function finds a data point in the queries and places it on the ‘main’ tab.



It is designed to be reused to populate the main tab with multiple data points. The pullSummary() sub-procedure calls pull() several times. It looks repetitive. I thought about creating a more elegant structure, but couldn’t justify the time it would take.

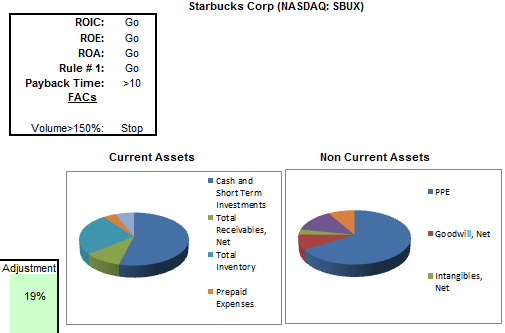


Turning Public Data Into Private Knowledge

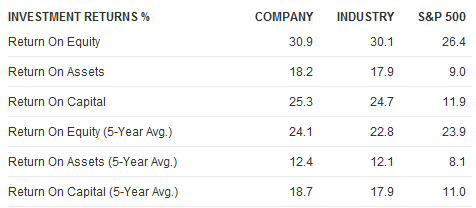
The primary purpose of this tool is to articulate meaning with the data. It does this in four ways:

* It only shows essential data on the main tab.
* It organizes tables more efficiently.
* It converts raw data into ratios that convey meaning.
* It consolidates balance sheet data into charts.

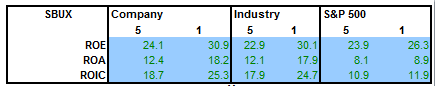
Front and center there are two pie-charts that depict the company’s assets in addition to a “Go List”. This list provides an immediate decision on the company. For example, the Go List for Star Bucks, SBUX, is positive. It indicates Starbucks is attractive and we should now wait for trading volume to pick up to enter the market. Inside five seconds I have a clear opinion that I like this company and I want to look at it further.



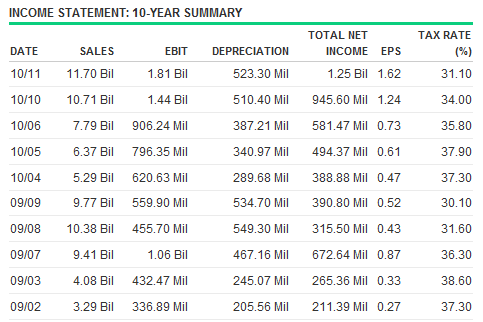
If I were to look up investment returns on MSN Money it would display the following table. To discover a trend my eyes spend a lot of time bouncing up and down the table comparing 5 and 1 year returns. I also waste a lot of time verifying that I am comparing the right numbers.



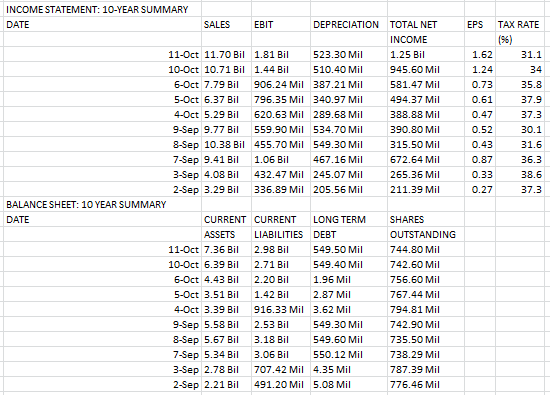
The equivalent table on my main tab allows me to make comparisons quickly and efficiently. I only have to look at the table for a few seconds to draw the conclusion that Starbucks is outperforming. This is an example of the pull() function at work.



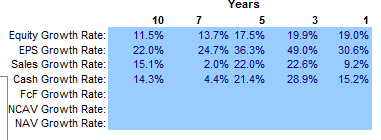
Some of the raw data needs more work to convey meaning. Below is a portion of the 10 Year Summary of the Income Statement.



It doesn’t look much different in the excel query.



These tables might be meaningful to a seasoned professional, but they are not for me. I used the raw data to create this table which easily conveys growth trends.



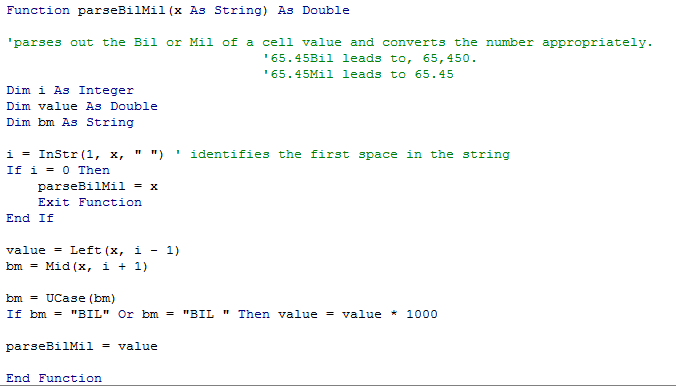
In the same two-tier modular structure pullFundamentals() calls the rate() function to process pairs of data points and place the results into the new table.

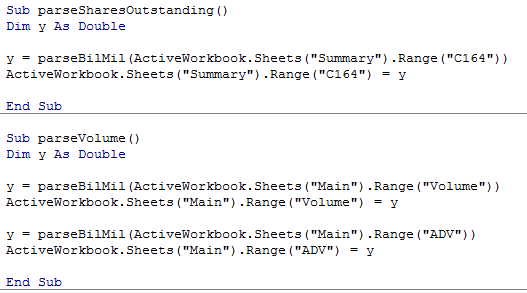
Parsing Volume & Share Data

Share and volume data require the “Bil” or “Mil” units in the cell value to be parsed out and the values converted appropriately, for example:

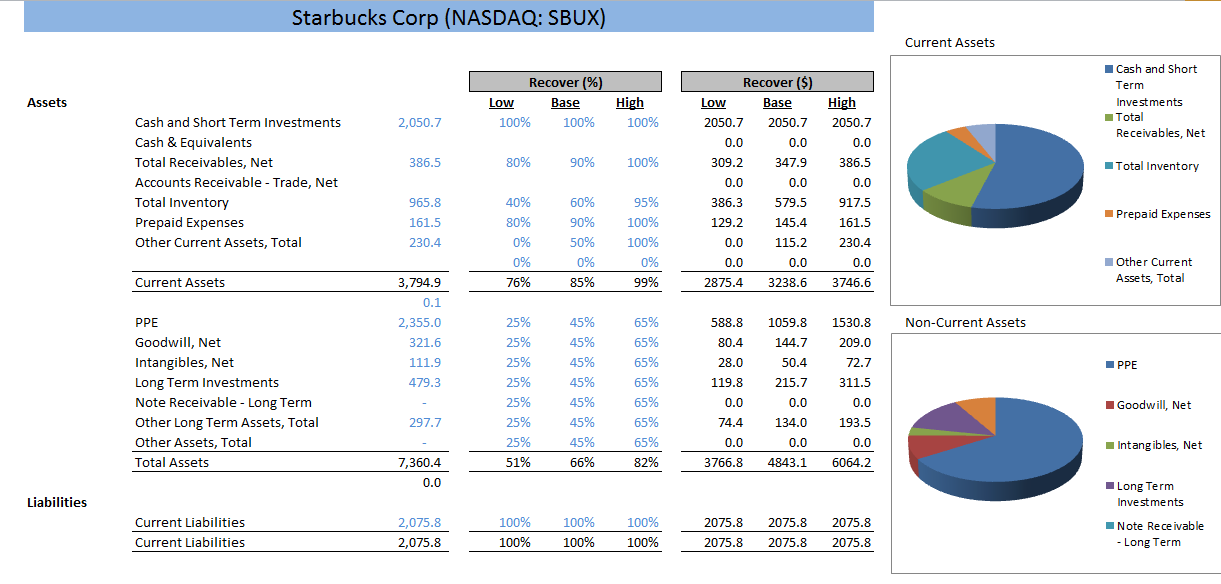
* + - * 65.45Bil must be converted to, 65,450.
      * 65.45Mil must be converted to, 65.45

The parseBilMil() function will parse these cell value and the parseSharesOutstanding() & parseVolume() procedures call parseBilMil() to parse their respective values. These sub-procedures are shown below.





Balance Sheet

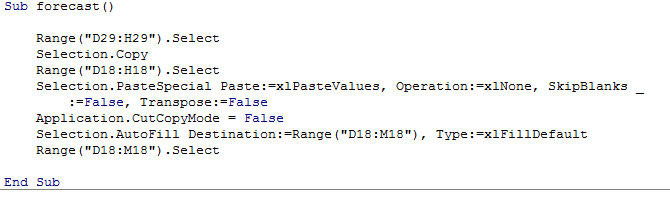


This tool uses annual data, but it pulls the most recent quarterly balance sheet using the quarterlyBS() sub-procedure. This data is populated into an acquisition-breakup (net-net) framework and is valued accordingly. This data populates dynamic charts on the ‘net-net’ tab as well as the ‘main’ tab.

Forecasting

In valuing a company I use discounted cash-flow models to estimate the value of the company. This is different from the PE based model that Town espouses in his book. I calculates the present value of future cash flows using an absolute 15% discount rate. There aren’t a lot of moving parts except for estimating what those future cash flows will be.

This model uses regression analysis to forecast the next five years of cash flows. While statistically rigorous it is a built-in function of the autocomplete feature and is trivial to execute. I recorded a macro and simply pasted the values and used autocomplete to generate the next five values. The computer did the rest.



Calculating a Pay Back Price

Town’s payback price is a theoretical ratio for most of us. It becomes realistic if we are thinking about buying an entire company, like Buffet. Suppose you bought the entire company at the current share price, how many years would it take to recoup your investment from the from the company’s earnings? The right answer is under ten. It isn’t an absolute rule, but it gives you another perspective for a name that is at an attractive price. Starbucks might be priced attractively, but according to is earnings it will be a while before I could recoup my investment. This framework helps me to think like an owner and not a gambler.

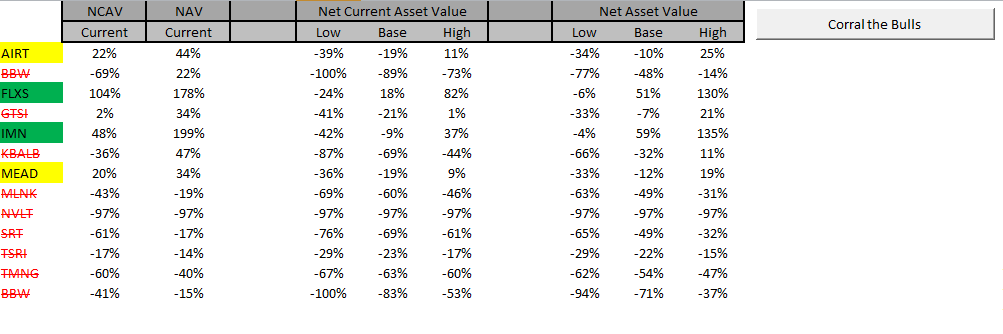
I thought about this code for a while. There are several ways you can go about it. I like to see what prices a name would need to trade at to get a payback time of 6 year or so. The comparison gives prospective on how attractively priced a name really is.

After thinking for some time, I decided that return for my time was becoming an issue and I decided on code that would run. This code runs. It starts at the first cell and works its way down (offsetting) in a loop until if finds a value that is higher than the current price. Once it does it offsets to the left and collects the value. It terminates at the 10th line. I am not interested in knowing beyond that.



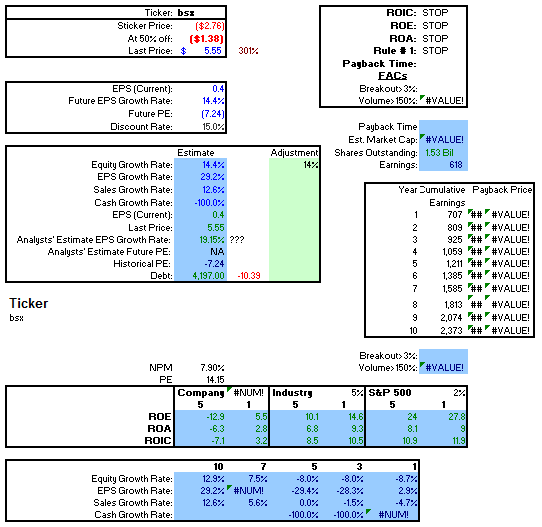
Corral the Bulls

The net current asset model is a basic balance sheet model that looks at the assets the company holds. Small companies sometimes trade for prices that are lower than their current assets, making them attractive. This semester, a class I supervise was assigned to research a set of net-net opportunities. I had to grade their presentations. I coded this the night before their presentations. I tested it on three companies until I was confident in the code. I ran it just before I went to sleep. I woke up the next morning and was pleased to have it during the presentations.

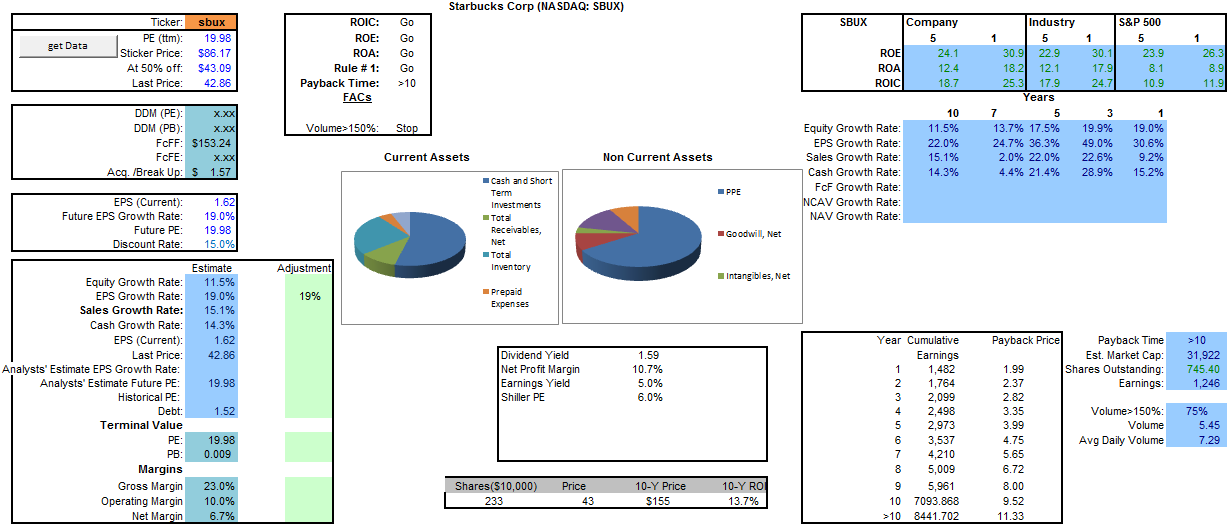


Getting the Data

To get the data and run the entire program the user updates an orange cell with the appropriate ticker and then clicks the getData button. This launches a macro that edits a succession of 20 queries from the MSN Money site. These queries are organized on a series of tabs that follow the MSN Money organization: Summary, Fundamentals, IE, CFS, BS, & Financial 10Y Summary. This works well, but after working with the data for several months I have learned that MSN’s data is not trustworthy. This is a problem I will tackle in the next iteration of the project. It works well enough for now. I can still use it as I develop the front end.

Lessons Learned

This screenshot shows the first iteration of this project in it’s entirety. It is comprised of excel logic, vlookups and automated WebQueries. At the time I didn’t know how to parse out information contained in a cell. You can see the ‘#VALUE! errors in the screenshot. These result from the ‘1.53 Bil’ value for shares outstanding. I had to parse this manually with each search. Once I did this the Payback Price table would update the prices. I also scanned the payback price table for the payback price because I didn’t have the programming skills to automate this task and populate the value into the ‘Go List’. Notice there isn’t even a button to run the program. I had to use a hot key. The first iteration provides a benchmark of what I could and could not do on the first day of class.



This is a screenshot of the second iteration. I learned a lot during this project. Using the first iteration as a benchmark, it would be accurate to say I learned how to do everything else. I seemed to pick up speed as the semester progressed. I wouldn’t say I am a fluent coder, but I am coding faster now than I did even two months ago. I struggled with loops and passing arguments (java) this summer; in these areas I have made successful progress.

I am not pleased with the DCF forecasting methodology. I want to run with it for a while and make observations. It was never intended as a serious solution, only a starting point.

The hardest part of the project was getting the right data from the queries. Each company has slightly different financials which makes this process testy. I was able to overcome this using find and vlookups.