

Dan Nebeker

MBA 614

4/12/2016

Variance Analysis - VBA Final Project

2.1 Executive Summary

The Variance Analysis Automation project that I've completed is one that will be of great value to the company where I am currently headed to work – Savage Services. Savage Services is an international supply chain services provider that offers logistics, transportation, loading, unloading and other supply chain services. I was an intern there last summer and am currently working part-time.

Savage manages hundreds of operational sites around the US, Canada and Middle East where they move rail cars, trucks, conveyor belts, pipelines, ships, barges, vessels, etc. Each operation must be financially monitored in order to succeed. The company hires business managers to keep a close eye on the finances on each of these hundreds of operation sites and monitor how closely they are meeting projections and budgets. Projection is basically a short-term (monthly) estimate of performance and budget is a reference to long-term (annual) estimate of performance. The difference between the actual results estimates are called variances. Savage employs approximately 70 business managers to monitor these variances. The financial data is pulled down from an internal system and follows the exact same format for every operation. The data is pulled down and analyzed daily by some managers who high volume operations and require to know every day if those variances are out of control. Other operations pull the data 2 or 3 times per week to track performance.

My Variance Analysis Automation tool works with the data pulled directly from our database for any of the aforementioned operations. Upon activating the sub procedure the macro will do some simple calculations of the downloaded data in order to obtain the necessary variances for analysis. After calculating the variances, a userform will pop up and ask the business manager for inputs that will define the limit of variances that should be analyzed. Probably because these limits vary from operation to operation and from manager to manager there has not been an internal system built already to handle this variance analysis. After the inputs are received through the userform, the procedure will identify variances outside of those controls and output a summary box and graphical illustration of the most consequential variances according to the userform inputs given. The average time saved each time a business manager runs the variance analysis is 20 minutes.

I know you could do the math to ballpark how much time and money this will save the company, but I will do it for you. If the data is analyzed on average 2.5 times per week by 70 business managers for 350 operations around the world who are on average making \$45 per hour, the total annual dollars saved as result of this project translates to:

Number of Operations for which the macro is used	350
Average number of times used per week	2.5
Hours of work saved per occurrence	0.3
<u>Average hourly rate</u>	<u>\$40</u>
Total Annual Company Savings	\$546,000

2.2 Implementation Documentation

Sub Procedure Variance

The sub procedure starts by defining some ranges that will be referred to later on in the code. The worksheet that is pulled in the form of an income statement and it is standardized so all of the necessary data is always in the same position. I used relative references offset by designated number of columns to run through 2 ranges and calculate the variances. One range is a calculation for the variance of actual to projection (short-term estimate) and the other variance is to the budget figure (a long-term estimate). There are 2 different conditions that must be considered when calculating the variances:

- 1) Revenue, Margin and Income Line Items- the variance is positive (good) if the actual figure is higher than the projected figure.
- 2) Expense Line Items- the variance is negative (good) if the actual amount turns out to be less than what was projected.

The 2 cases are tricky for a manager to decipher between a good and bad variance and therefore I used conditional formatting to expressly illustrate when a variance is good or bad. Good variances are formatted with the property Style = "Good" for every cell in the defined range using a For Each Loop for both conditions.

The subprocedure formats the new columns containing the variances to match the format of the existing financial statement. Specifically, I used the following properties to format the various ranges and other objects:

- 1) Font.Size
- 2) Font.Bold
- 3) Borders.Weight
- 4) Borders.LineStyle
- 5) Alignments
- 6) AutoFit

For much of the formatting, I recorded macros and then used the recorded code to identify color names, border styles, etc. so that the output columns exactly match the standard statement.

```
'conditional formatting

'set the formatting for revenue type cells
For Each revCell In revItems

    If revCell.Value >= 0 Then
        revCell.Style = "Good"
    Else
        revCell.Style = "Bad"
    End If
Next revCell

'set the formatting for expense type cells
For Each expCell In expItems

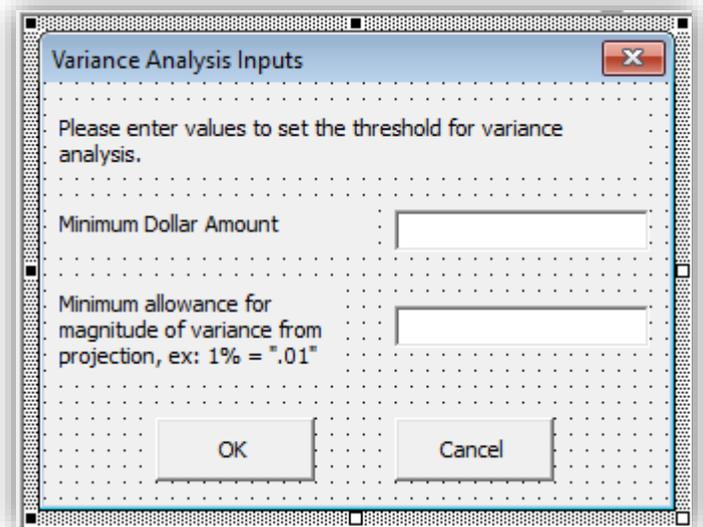
    If expCell.Value <= 0 Then
        expCell.Style = "Good"
    Else
        expCell.Style = "Bad"
    End If
Next expCell
```

The last thing that the sub procedure does after calculating and outputting variances is it invokes and shows a userform to collect inputs from the business manager.

Userform VarianceInputs

I inserted a userform that collects 2 pieces of data from user:

- 1) Minimum Dollar Amount- This sets the minimum dollar of variance that should be identified by the business manager.
- 2) Magnitude of Variance- This should be entered as a percent and will only accept a number between 0 and 1



Upon clicking OK, the inputs are received and they are assigned to variables and used to make calculations and build a summary of valuable information for the user. Cancel is set up as a cancel button and will hide the form and exit the sub procedure.

After clicking OK, the code uses a Do Until Loop with a counter to cycle through all 99 line items and determine if it is one that meets the parameter specified by the business manager through the userform. Inside of the Do Until Loop, the first thing I do is use a Boolean variable to tell the program whether or not the line item that is being analyzed is a Revenue/Income item (True) or an Expense Item (False). Then, I used a series of nested If, Elself statements to allow the program to decide which values should go into the summary box and in what form- good or bad variance.

Scenario 1) This scenario occurs when the actual estimate and the projected estimate are zero. In this scenario, the code should simply move on to the next line item. No calculation necessary and no output to the summary box.

Scenario 2) This scenario occurs when there is an actual number but nothing projected. In this case, any revenues are a good variance and any expenses are a bad variance. And depending on whether or not the dollar amount is over the user defined limit, this will be output to the summary box.

Scenario 3) This scenario is the opposite of scenario number 2. A projection was made but there is no actual figure. Same output as scenario 2.

Scenario 4) This is the catch-all statement that occurs most frequently. It occurs when there is a projection (that is not 0) and an actual number (not equal to 0).

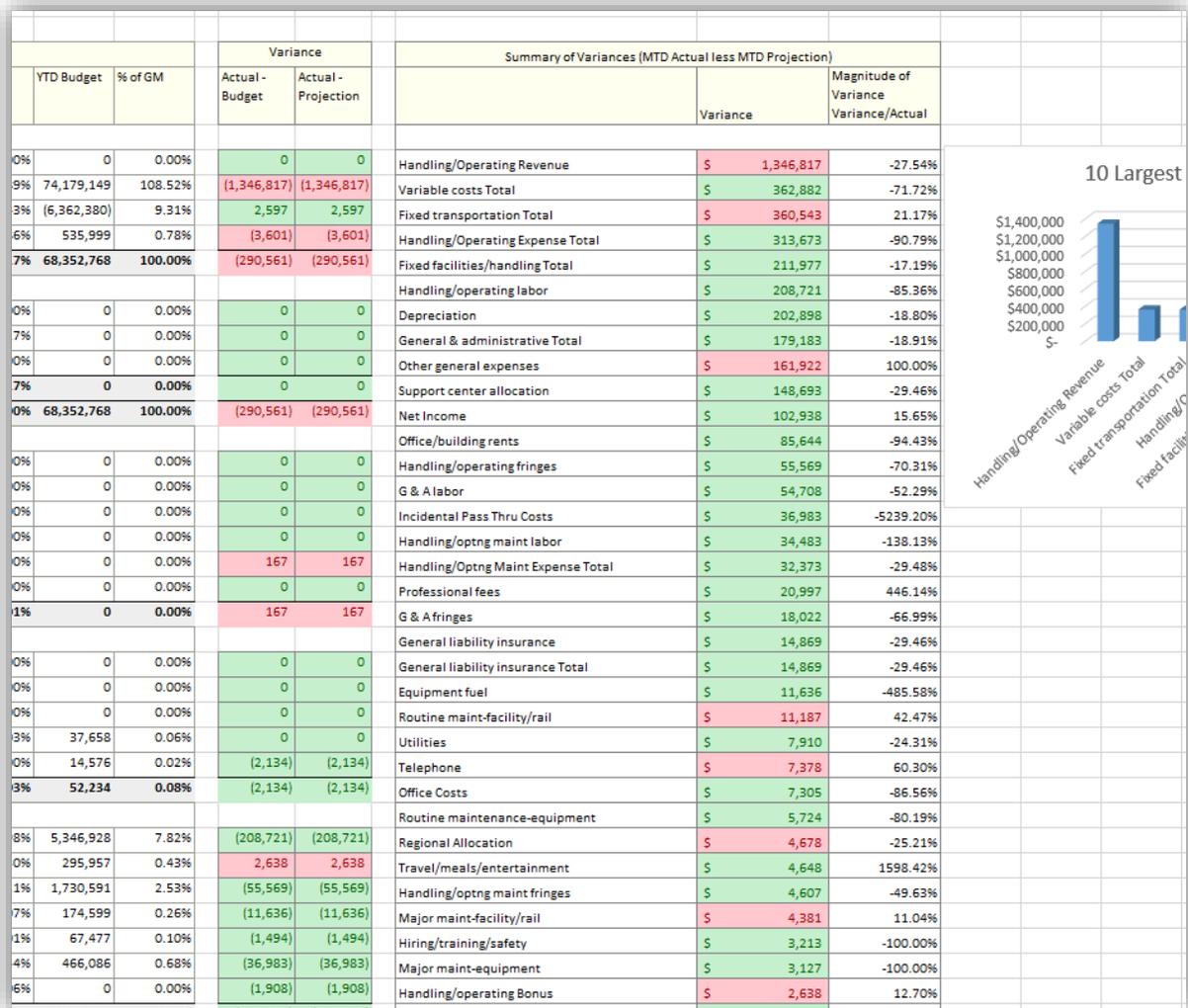
Now that all of the significant (as defined by each individual business manager) have been output to a new range of cells and conditionally formatted accordingly, that new range is defined using the counters utilized in the do until loop so that it can be manipulated and used for the rest of the analysis.

Now, the variances in the summary box are assigned to a range and that range is turned into absolute values using a For Each Next Loop. Once the values are absolutes, they can be sorted nicely.

Next, the entire summary box of variances and their associated magnitudes of variance from the actual number is sorted in the program by calling the sort method and using the newly defined summaryBox Range. This essentially ranks the value of importance and significance of the variance for the business manager.

Finally, the nicely arranged summary box of valuable information is illustrated with a bar chart by taking the 10 most significant values (values arranged in the top 10 rows of the summary box). The bar chart is positioned nicely next to the summary box.

The summary box is also formatted to match the style of the downloadable report. Titles are made using bold and interiorcolor properties. Font sizes are adjusted to match. Number formats are arranged to display dollars without cents and magnitudes are expressed as percentages. The following screenshot displays the original income statement and some of the adjustments including the variance outputs, summary box and bar chart all formatted nicely and matching the original report.



2.4 Discussion of Learning and Conceptual Difficulties

The first challenge associated with the project was thinking freely along the lines of what problem do I want to solve. I wanted to create something useful. I asked some of the business managers which whom I work what tasks they do daily, weekly or regularly that are redundant. Each manager replied that variance analysis is redundant and time consuming. This led me down the road of trying to automate a solution.

This is the first real project that I've undertaken in VBA that doesn't have a well-defined list of instructions. Therefore, I quickly learned how powerful the recorder is in helping give clues on how to perform a task. On a few lucky occasions I could record the task, copy the recording and paste it right into my code. Most often I could only use the recorded code as a guide on how to perform the task. Nevertheless, it was very helpful throughout the project.

The first obstacle I hit was conditional formatting the variances to be good or bad. I came across the pre-defined property styles "Good" and "Bad" and they worked nicely once installed.

I really had to ask myself what kind of loop to use to cycle through the line items. Throughout the project I used the Do Until Loop and the For Each Next loop to repeat tasks. I learned the benefits associated with each and how to manipulate each style to target a specific range.

My ability to work with ranges greatly increased during the project and as I advanced into later stages of the project I became much more comfortable defining ranges and calling them by their name rather than simply a line full of cell numbers.

This project allowed me to see how to nest if statements inside of loops in order to capture conditions for different scenarios depending on the variables that feed into the loop. I found myself using debugging tools that baffled me at the beginning of the semester. I would ask myself, why is Gove doing this weird debug.print thing but during this project I found myself spitting out debug.print lines regularly to check values and help my brain stay in sync with the computer as it runs through the code line by line.

One concept that I had difficulty overcoming was how to run 2 counters simultaneously while in a Do Until Loop. The first counter ensured that the program works through every line item on the income statement. The second counter was building a summary box of information who's size is dynamic and will be used later in the code to format and manipulate. Once I got the counters to work together in sync, I was impressed by the power of VBA.

The proper usage of a Boolean variable jumped out to me when I was trying to figure out how to designate which type- Revenue or Expense- each line item should be assigned. Using the Boolean variable allowed me to eliminate huge blocks of unnecessary code, mainly composed of if then statements. The following screenshot illustrates the usage of the Boolean variable *isRev* to determine what type of variance and therefore which scenario to enter and illustrates usage of the 2 simultaneous counter variables: *sumRow* for the summary box counter and *r* for the line item counter.

```

r = 10
sumRow = 10
Do Until r = 99
    If r = 10 Or r = 11 Or r = 12 Or r = 13 Or r = 14 Or r = 20 Or r = 58 Or r = 59 Or r = 60 Or r = 61 Or r = 62 Or r = 63 Or r = 64 Or r = 65 Or r = 66 Or r = 67 Or r = 68 Or r = 69 Or r = 70 Or r = 71 Or r = 72 Or r = 73 Or r = 74 Or r = 75 Or r = 76 Or r = 77 Or r = 78 Or r = 79 Or r = 80 Or r = 81 Or r = 82 Or r = 83 Or r = 84 Or r = 85 Or r = 86 Or r = 87 Or r = 88 Or r = 89 Or r = 90 Or r = 91 Or r = 92 Or r = 93 Or r = 94 Or r = 95 Or r = 96 Or r = 97 Or r = 98 Or r = 99 Then
        isRev = True
    End If

    If Cells(r, 13).Value = 0 And Cells(r, 17).Value = 0 Then 'if actual and prod are both 0
        r = r + 1
        'move on

    ElseIf Cells(r, 13).Value = 0 And Cells(r, 17).Value <> 0 Then 'if actual is 0 and prod is not 0
        magnitude = -1
        variance = Cells(r, 25).Value
        If Abs(magnitude) >= percVar And Abs(variance) >= dollarMin Then
            Cells(sumRow, 28).Value = Cells(r, 1).Value
            Cells(sumRow, 29).Value = variance

            If isRev = True And variance >= 0 Then
                Cells(sumRow, 29).Style = "Good"
            Else
                If isRev = False And variance <= 0 Then
                    Cells(sumRow, 29).Style = "Good"
                Else
                    Cells(sumRow, 29).Style = "Bad"
                End If
            End If
        End If

        Cells(sumRow, 30).Value = magnitude
        sumRow = sumRow + 1

    End If
    r = r + 1
End Do

```

For the project, I tried for a couple of hours to automate the process of reaching out to our company's database through the web portal in order to download the data right through excel. There were security issues I was bumping into multiple layers of inputs that needed to be submitted (login credentials, operation, report type – daily, monthly, annual) etc. so I gave up on that attempt and require the user to find the report online and then simply use the built-in export functionality to download the report that needs to be analyzed. I saved my sub procedure at the Personal Level macro which allows me (and other business mangers) to use it for all of the operational sites in the company. I can download a new report for any operation site and simply run the macro and have my variances all laid out in front of me in less than 10 seconds.

2.5 Assistance

I received no help from anybody on this project other than random people who have contributed random blocks of VBA code to the World Wide Web.