Final Project

BYU - MBA 614

Professor Gove Allen

Shopify Sales Data Analysis Macro

For

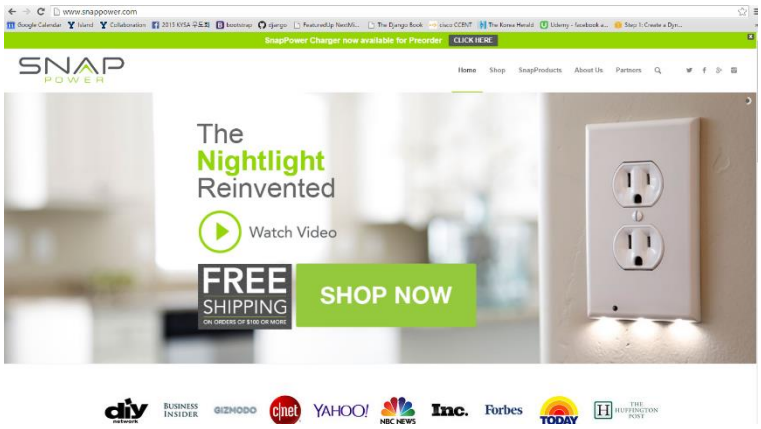
SnapPower

Jong Park

4.13.2015

Executive Summary

Description of business



SnapPower is a local-startup company that sells built-in-LED-Guidelights as electrical outlets on walls. It has been about a year and a half since the company went live. It has been about a year since they have started to sell online using Shopify as their E-commerce platform.

The company sells 6 different types of Guidelights, with 6 different packaging types. This equals to 36 possible 'product types' that the customer can place.

Currently, the company is not doing any analysis on the products that have been sold. For the last twelve months, the company has been tracking the number of orders per week and the corresponding revenue for them, but there has not been a system put in place to track the orders by type, packaging, or days.

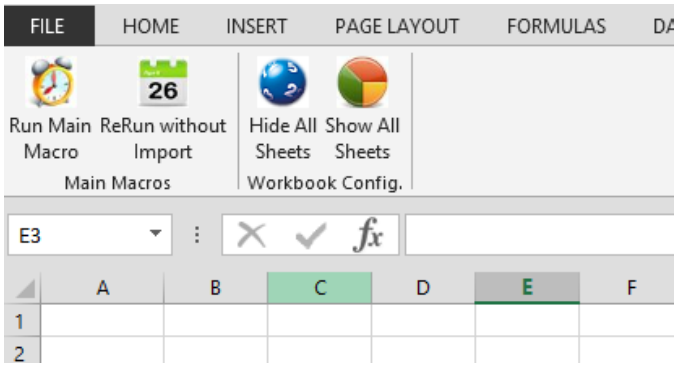
The General overview of the system

The System that I built provides the user with various statistics that were not kept track of previously. They include orders by type, packaging, days, weeks, and so forth. These additional statistics will help the company analyze trends to set marketing strategies in the future.

A user manually downloads a csv file from the Shopify website. After that, the macro starts off by prompting the user to select the downloaded file. The sheet in the raw data file is copied over to the project file and the days and weeks are filled in according to the dates that the raw data uses. It then goes through and organizes the data by the specified fields.

Implementation

The Workbook Ribbons



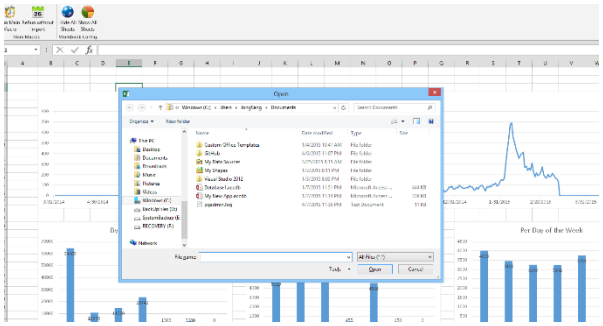
There are four buttons on the ribbon tab labeled "information". The functionalities of the four buttons are as follows:

- "Run Main Macro": This button will run the main macro. The main macro first prompts the user to import a downloaded Shopify raw data file that the user receives daily through their email. If the user cancels, the macro stops altogether. When the user selects a downloaded file to import, the macro opens the specified file, copies over the main sheet to our project file, and executes the analysis through the macro. More details will be explained in the following sections.
- "Rerun without Import: Achieves the same thing without importing a new downloaded file. This is used when the user wants to modify the raw data file that already exists in the project file and rerun the analysis.
- "Hide All Sheets": Hides all the raw data and analysis sheets so that the user only sees the sheet with the summarized charts.
- "Show All Sheets": Shows all the sheets that the user can analyze further.

The Macro

This section includes descriptions of the sequence of the macro. Samples of the 'technologies' or methods used in each section are bulleted.

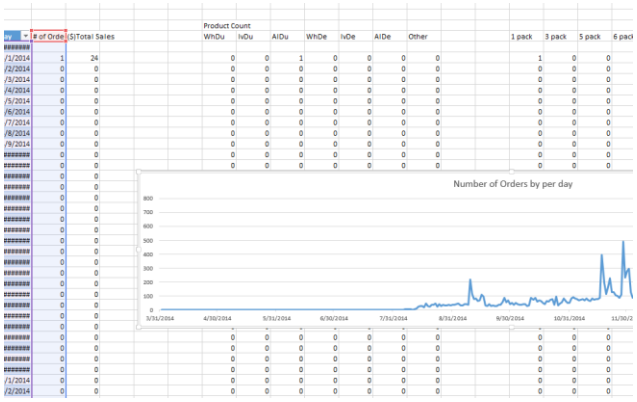
The sequence of the macro is as follows:



1. **The user is prompted to import a downloaded csv Shopify sales data.**
    a. The downloaded workbook is opened. The sheet(1) is copied over to the project file.

- Application.GetOpenFilename()
- Workbooks.Open

2. **The project workbook keeps a copy (replaces the old sheet with the copied sheet) of the csv file and closes the csv.**
   a. The existing raw data sheet in the project file is replaced with the new sheet. Then, the downloaded file that had the raw data is closed without save.
      - Application.DisplayAlerts = False
      - Workbooks(2).Close SaveChanges:=False
3. **Prepares the project workbook to begin analysis. This process includes:**
   - Retrieving the list of product names as SKU's from the raw data sheet to the 'ProductSheet' so that it has a reference to plug in 'SKU's back into the raw data in case there are missing entries.
     - Application.WorksheetFunction.VLookup(Cells(y, Lineitemcolumn).Value
   - Getting a range of dates in the raw data and putting that into the 'ByDay' and 'ByWeek' sheets so that the summarized data doesn't include any unnecessary dates.
     - MostCurrentDate = Month(MostCurrentDateTime) & "/" & Day(MostCurrentDateTime) & "/" & Year(MostCurrentDateTime)
     - Select Case Weekday(MostPastDate)
4. **Analyzes by dates and plugs the information into the 'ByDay' Sheet.**
   a. A single object of the order class is then instantiated. The idea is that all of the orders for each day are combined and summed up into one object. For example, one line on the raw data has a date of 2000.1.1 and an order of product1. The object's variable for product1 would now be 1. Then, the next line also has a date of 2000.1.1 and an order of

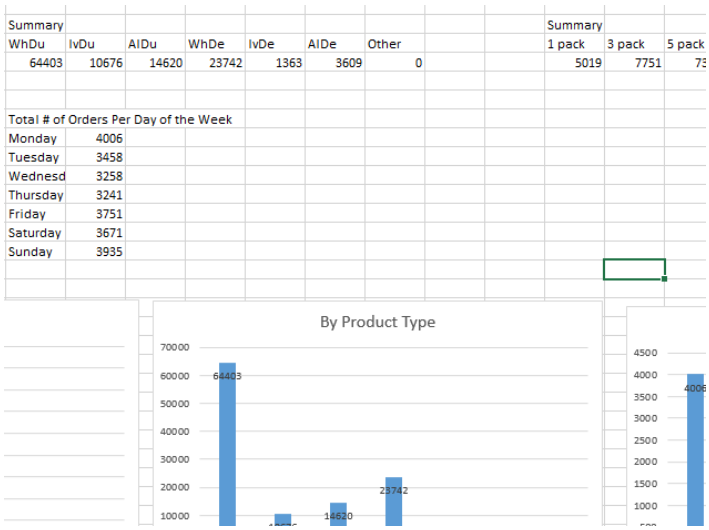


product1 as well. The object's variable for product1 would now be 2. It will keep adding until the date changes, which at the point it will empty out its values on the corresponding date on the 'ByDay' sheet and the object will have variable values = 0. Then the process is repeated.

- Class "Order"
- Select Case productname

```
Case "01wh-101"
  daysOrders.WhDuPack1 = daysOrders.WhDuPack1 + 1 'white duplex
Case "03wh-101"
  daysOrders.WhDuPack3 = daysOrders.WhDuPack3 + 1
…
```

5. **Analyzes by weeks and plugs the information into the 'ByWeek' Sheet.**
   a. Same process as above, but on a weekly basis.
6. **Summarizes the data in the 'ByDay' Sheet (sums, by week day, etc.) at the bottom of the 'ByDay' Sheet.** It makes use of worksheetfunctions.



Summary										Summary		
WhDu	IvDu	AlDu	WhDe	IvDe	AlDe	Other				1 pack	3 pack	5 pack
64403	10676	14620	23742	1363	3609	0				5019	7751	7

Total # of Orders Per Day of the Week	
Monday	4006
Tuesday	3458
Wednesd	3258
Thursday	3241
Friday	3751
Saturday	3671
Sunday	3935

- Application.WorksheetFunction.Sum(Range(Cells(5, m), Cells(sumrow - 1, m)))

7. **Generates 10 charts based on that data and positions copies of them in the 'Summary' sheet.**
   - ActiveSheet.Shapes.AddChart2(227, xlLine).Select
   - Selection.Top = oCell.Top
8. **Hides all the sheets except for the summary.**
   - Sheets("ProductSheet").Visible = False

Discussion of Learning and Difficulties

A couple parts of the project were challenging, as I did not have experiences in the topics.

The first problem I faced was how to design the project before I began. Each product type required a separate variable, which meant there were 36 separate variables to begin with. I ended up using a class to create an object for each sale. Each object would have a date with all the 36 variables associated with it. It was a bit of thinking to do, but it worked out quite simply.

The second issue was working with charts. The charts are treated on the same level as the sheets – as workbook objects. Therefore, it was difficult to get the code working to copy over the actual charts to the summary sheet. What I did instead was I copied them over as images to the summary sheet, but I kept the original charts in the data sheets so that the user can reference, update, and alter the tables at will.

Substantial Assistance

- I have not received any assistance in the making of this project other than stackoverflow for snippets of code.