Scott Walker
Thomas Kelly
Prof. Gove Allen
ISYS 540

Write-up for Our Project

Executive Summary of the Project

Description of the Business

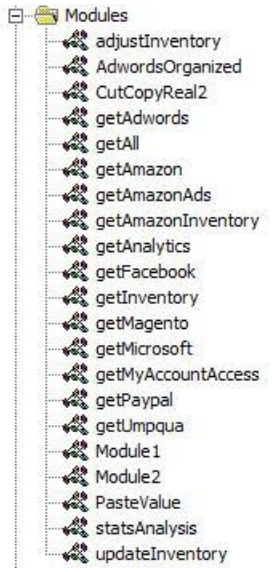
Underwater Audio is a business which waterproofs iPod Shuffles. Their main activity is buying the iPods themselves and waterproofing them (though an option to send in your iPod is available) then selling those with waterproof headphones. They sell their products in retail stores, on their own website, and through amazon. In order to determine their profit we must consider their costs. Their main cost is advertising - they spend money on Facebook, Google, and Amazon. All their current cash is also in differing locations. Namely their bank of choice, MyAccountAccess, Amazon, and Paypal.

Overview of the System We Built

This excel file uses Professor Gove's internet scrapper to login to all necessary accounts (listed in the previous paragraph), go to the current day, and download the current day's data in each account to their respective tabs. All of the sites involve a login, and many have security features. The essential data to provide in analysis is then both found (providing flexibility for the data to move around a bit) and copied to one tab. Which is then inputted into the massive worksheet which contains all of the data. A graph has been made which is both flexible and dynamic this was made using extensive research and involves named ranges. The graph also can compare whichever of the fields you choose giving a R2 value -- this allows for actual correlations to be found and business decisions to be made.

Implementation Documentation

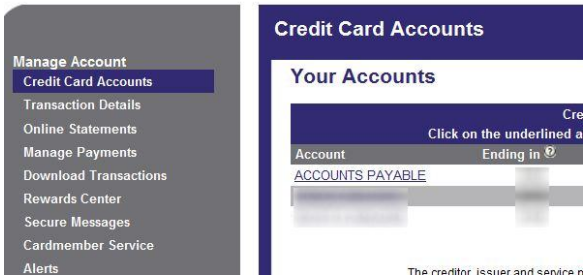
Documentation of What We Actually Did For Our Solution.



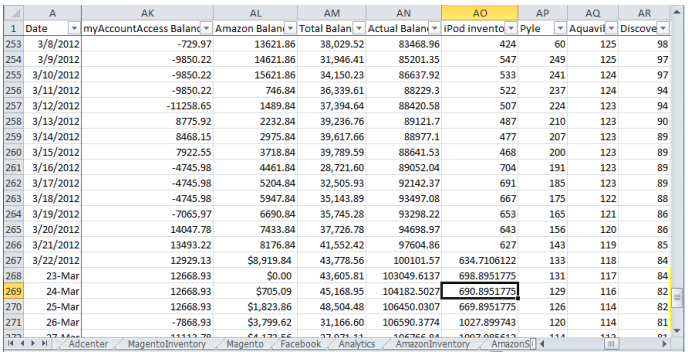
We went to work based on our list of necessities and one by one completed the web scrappers. We made many modules each named “Get” and then the key name of the website scrapped. With one module we called all the gets. For each we used the scrapper agent provided by Professor Gove. Whenever I say that we logged in to a page it required looking for the tag inline just before the inputs for username and passwords then using their id’s or sometimes some other feature to input the correct username and password. And then finding and clicking the “go” button.

For “getAmazon” first it opens internet explorer and logs in. Then it deletes the Amazon tab and recreates it anew. Because there is a bit of information necessary from that first page it is imported then. Then by having in our possession the website that it’s necessary to have for the next set of information and already being logged in we open another page on amazon and imports that page to AmazonSales. For “getAmazonAds” first it opens internet explorer and logs in then since the web page is simple and known it is opened and imported to AmazonAds. For “getAmazonInventory” first it opens internet explorer and logs in. Then similar to the above it logs into amazon and imports the page to AmazonInventory. For “getAnalytics” first it opens internet explorer and logs in through a specific webpage that makes it simple to get to the information necessary. Then we added code to wait a bit for the complex page to load. Then the page is imported to a tab named Analytics For “getFacebook” first it opens internet explorer and logs in using a specific website (getting us nearly to the data we could). We had to find and click several buttons. These buttons change the date to our target date and refresh the page. Then we imported the page. For “getInventory” first it opens internet explorer and logs in. This one deals with a google spreadsheet and after you arrive there importing it is the only other step. For “getMagento” first it opens internet explorer and logs in. At this point we need to find a button with the value submit and click

it. This page is then imported as the Magento tab. After a couple of more button clicks and page loads we reach another page with vital information. This is imported as the MagentoInventory tab. For "getMicrosoft" first it opens internet explorer and logs in. Then after finding the dateFiller div it pastes the correct dates using "Format(Date - 2, "m/d/yy")" into the inputs. A button is then clicked confirming the refresh and new dates. Then the information is imported. For "getMyAccoutAccess" first it opens internet explorer and logs in through a two part sign in.



After several clicks the information needed is imported. For "getPaypal" first it opens internet explorer and logs in. After a button click we are at the information required and we say import. For "getUmpqua" first it opens internet explorer and logs in through a two part sign in. Then after searching for a tag name and clicking a button we are on the page to import and this is done.



	A	AK	AL	AM	AN	AO	AP	AQ	AR
1	Date	myAccountAccess Balanc	Amazon Balanc	Total Balanc	Actual Balanc	iPod inventc	Pyle	Aquavil	Discove
253	3/8/2012	-729.97	13621.86	38,029.52	83468.96	424	60	125	98
254	3/9/2012	-9850.22	14621.86	31,946.41	85201.35	547	249	125	97
255	3/10/2012	-9850.22	15621.86	34,150.23	86637.92	533	241	124	97
256	3/11/2012	-9850.22	746.84	36,339.61	88229.3	522	237	124	94
257	3/12/2012	-11258.65	1489.84	37,394.64	88420.58	507	224	123	94
258	3/13/2012	8775.92	2232.84	39,236.76	89121.7	487	210	123	90
259	3/14/2012	8468.15	2975.84	39,617.66	88977.1	477	207	123	89
260	3/15/2012	7922.55	3718.84	39,789.59	88641.53	468	200	123	89
261	3/16/2012	-4745.98	4461.84	28,721.60	89052.04	704	191	123	89
262	3/17/2012	-4745.98	5204.84	32,505.93	92142.37	691	185	123	89
263	3/18/2012	-4745.98	5947.84	35,143.89	93497.08	667	175	122	88
264	3/19/2012	-7065.97	6690.84	35,745.28	93298.22	653	165	121	86
265	3/20/2012	14047.78	7433.84	37,726.78	94698.97	643	156	120	86
266	3/21/2012	13493.22	8176.84	41,552.42	97604.86	627	143	119	85
267	3/22/2012	12929.13	$8,919.84	43,778.56	100101.57	634.7106122	133	118	84
268	23-Mar	12668.93	$0.00	43,605.81	103049.6137	698.8951775	131	117	84
269	24-Mar	12668.93	$705.09	45,168.95	104182.5027	690.8951775	129	116	82
270	25-Mar	12668.93	$1,823.86	48,504.48	106450.0307	669.8951775	126	114	82
271	26-Mar	-7868.93	$3,799.62	31,166.60	106590.3774	1027.899743	120	114	81

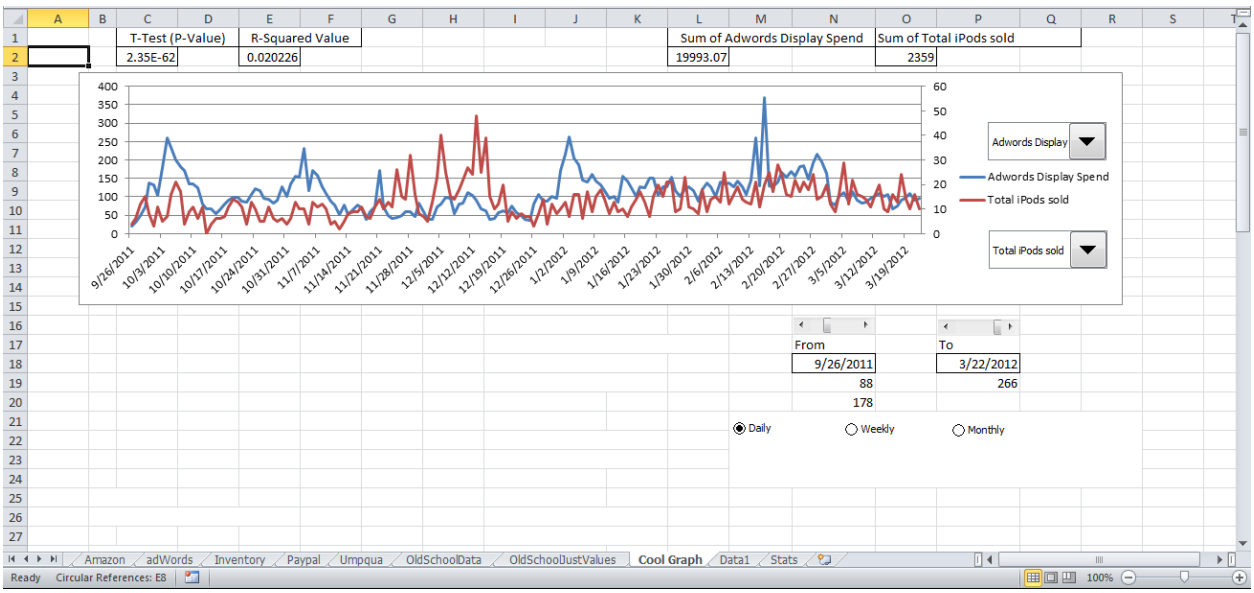
Adcenter | MagentoInventory | Magento | Facebook | Analytics | AmazonInventory | AmazonSl

Next a parser module was made to look through all the imported data, copy what was needed and paste those to a single tab. This is actually quite complicated and comes basically in two kinds of code. Both are made inherently flexible because they search for a key word and then relative to that cell move over to get the data necessary (this was absolutely necessary for one but was made true for both). The first kind was easiest where a simple cut and paste was required. Although some of them required a mathematical function to complete their work. The second kind were more difficult as a target date was necessary and comparisons to other worksheets had to be made. For

	B	C	D	E	F	G	H	I	J	K
32										
33										
34	Pending Transactions (Pending amounts may not represent final transaction amount)									
35										
36	Date			No	Type	Description	Category	Debit	Credit	Inquiry
37	4/2/2012				Credit	PROCESSING:			$2,182.52	$22,762.88
38	4/2/2012				Credit	PROCESSING:			$1,788.75	$20,974.13
39	4/2/2012				Credit	PROCESSING:			$2,514.96	$18,459.17
40	4/2/2012				Debit	PROCESSING:		$142.98		$18,602.15
41	4/1/2012				Debit	PROCESSING:		$47.12		$18,649.27
42	4/1/2012				Debit	PROCESSING:		$98.56		$18,747.83
43	3/31/2012				Debit	PROCESSING:		$93.99		$18,841.82
44	3/31/2012				Debit	PROCESSING:		$116.44		$18,958.26
45	3/31/2012				Debit	PROCESSING:		$20.99		$18,979.25
46	3/31/2012				Debit	PROCESSING:		$32.66		$19,011.91
47	3/31/2012				Debit	PROCESSING:		$16.84		$19,028.75
48	3/31/2012				Debit	PROCESSING:		$19.20		$19,047.95

example the “Ghost” accounts are there due to the report delay of anywhere from 2-5 days from any one money holder to the bank. So these types of data had to be created through comparison and dummy variables. After all the data is put in their respective tabs they are then copied and pasted to the mass enumerator of the data which is connected to the graph.

The Graph was a project in and of itself. Much research was put in to get this final product. This graph enables the user to dynamically change the columns of data to compare, the date range, and the type of dates to compare (daily, weekly, or monthly). The chart is created using named ranges, the Offset formula, and multi-layered If functions. With the flexibility afforded by this chart, I wrote a macro that enables the user to analyze the statistical relationships between every possible combination of column in the data for the date range and type of dates selected by the user. The macro then records the significant statistical relationships in the “Stats” sheet. This program enables the user to both discover relationships between variables and learn which variables are not significantly correlated. The statistical analysis macro can be run from the top ribbon. This graph and macro together give the user a powerful tool for discovering important relationships and actionable data.



Discussion of Learning and Conceptual Difficulties Encountered

There was an issue with the Google Analytics scrapper. It would not change the date as simply as the other scrappers had. It seemed we had to click a feature, input the dates, check and uncheck a box, and then click the button to get it to work. After we implemented this it still did not work because the inputted dates did not seem to register (there was a javascript tied button/calendar that needed to be clicked - which was seemingly impossible to place with the agent). Fortunately we pulled up an older version of the site which used a simpler feature in order to change the date and that's how it was solved, but it took some time to find that.

There were issues setting up the graph because originally I tried to create it without the use of named ranges. The named ranges referencing the data enable the graph to dynamically incorporate new data. Also, the date and date type features of the graph would be impossible without named ranges. We created the graph from several unrelated sets of instructions that forced us to reconcile the differences between them in creative ways. The date type functionality required additional formulas of our own creation. For instance, many of the columns of data are not running totals. This required us to create an If statement that both determined if the column was a running total and summed it depending on whether the data was supposed to be displayed daily, weekly, or monthly. We encountered several issues in creating the graph and statistical analysis macro that forced us to engineer our own creative solutions.

Assistance

This was a group project with Thomas Kelly and Scott Walker. We received no outside help barring advice from the Professor.

Write-up Detail

In Summary:

- We made a macro which scrapes information from 10 complex websites using an agent
    - This required different methods of finding ids, names, and dealing with situations where there was neither of those.
- Then we made a macro which scoures that imported information and is flexible (accounts for minor changes in how the information is imported).
    - The macros for this section sometimes simply copy and paste the found information.
    - Other times the macro must add any number of days (dates) worth of additions and subtractions to get to the target date.
    - And again other times this macro must compare a very specific piece of data on one sheet and compare it to something which may or may not be there on the other sheet.
- Then a graph was made to compare all this data. (including a macro to find statistical correlation between all of the variables)
    - The graph uses nested If functions, the offset functions, slider controls, combo boxes, and named ranges to enable the user to customize the columns to compare, the date range, and the date type (daily, weekly, monthly). The macro then uses the cells that control the combo boxes to compare every possible combination of columns and record the columns with a statistically significant correlation in the “Stats” tab.