John’s Copart Auction Scraper

By Zachary Blaszczyk

Executive Summary

This project was built for my dad's friend, John. He runs a small business where he purchases cars from auctions, fixes them up, and then sells them. John wanted me to build him a program where he could download the auction site's list of cars and then scrape the web for the cars rough resale value on the private market. The website I have decided to scrape through is Kijiji, the Canadian Version of KSL, and the auction site is Copart.

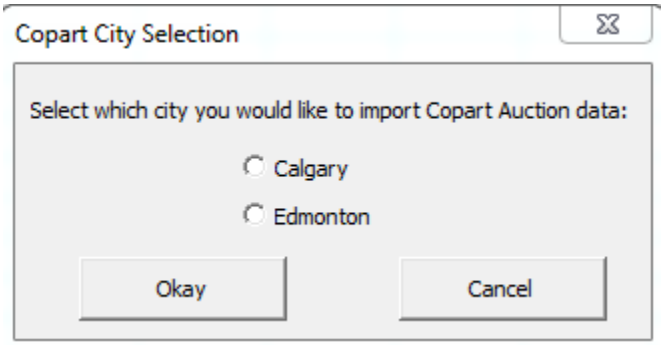
The program I built is really two parts: 1) getting the data from the auction site and 2) pricing each car available for auction from an online sales website.

Implementation

Below I will describe the major parts of the project and break it down as necessary.

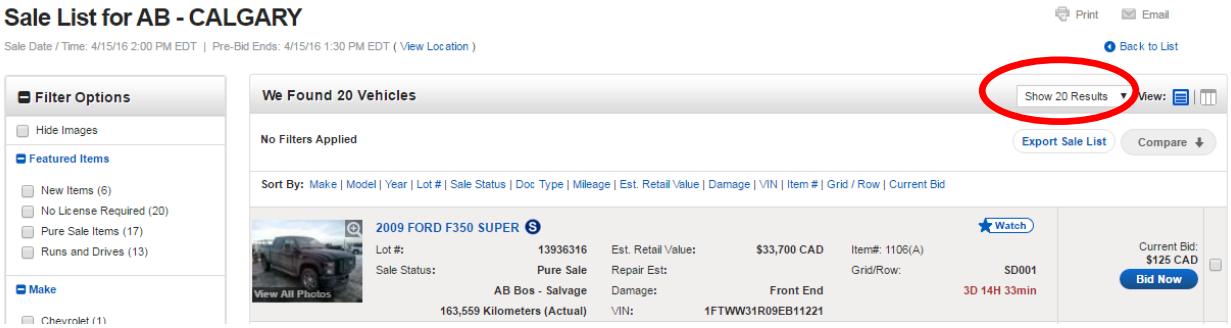
1. This project is designed to take listings off the car auction site, Copart, and market prices for the cars about to be auctioned from the website Kijiji. This project has two main parts to it, which can be seen by the tab added to the ribbon.

2. Due to the location of John's business, he would like to choose between two different

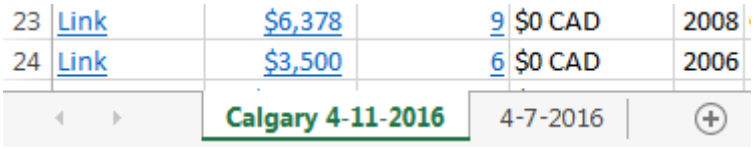


auctions located in two different cities. To allow him to do so, I created a user form (pictured below).

3. Once he selected the desired city, the program then went to the Copart website.
    a. Using the agent, the program sifts through the initial website to find the desired city's website.
    b. Once the desired city's website is up, I had to Google and find some code that could download a CSV file from the website. The CSV includes all the needed information about the cars that are going to be auctioned that week.



4. Manipulating the CSV file
    a. The next step after downloading the CSV file, was to then locate it and open it up.
    b. Once opened, I copied and pasted all the information into a new sheet created in the program. The new sheet is a combo of the name of the city in which the auction is taking place, and the date the data was pulled. Note that I didn't make a button that would update the current bid prices, this was because John wanted this program to be more of a reference tool. The actual bidding he does personally on the Copart website.



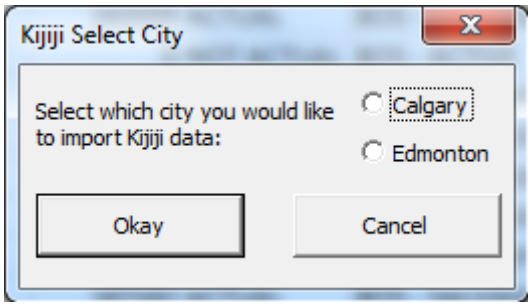
    c. The program then closes and deletes the CSV file. This is actually really slick because there is no trace of the old CSV file left on the computer, so the user doesn't have to worry about a huge folder full of useless CSV files being stored on their computer.
    d. Once the data was safely moved into a new sheet in my program I manipulated the data as to John's specifications, and I formatted the column headings.

Calgary Copart Import

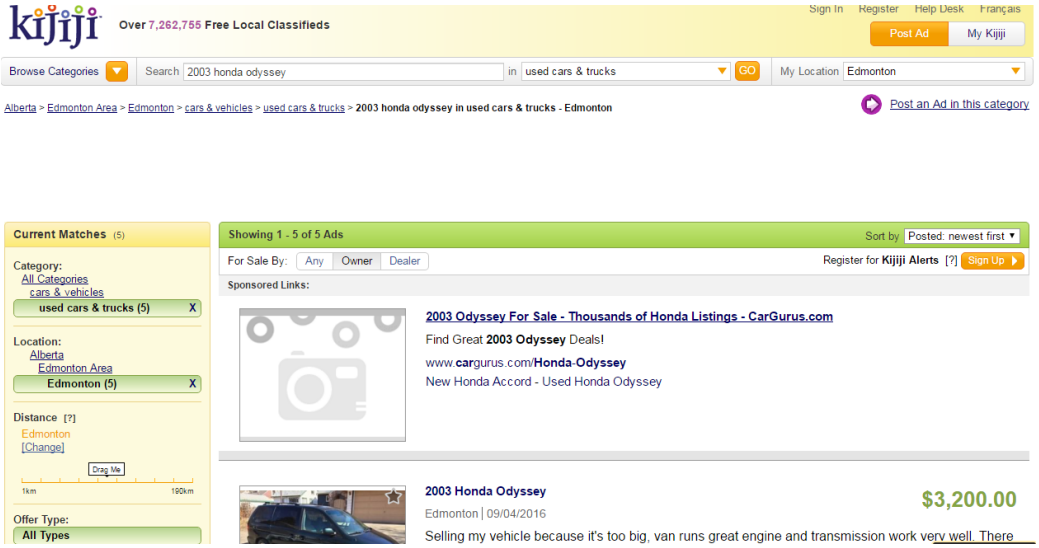
Imported: 4/11/2016

Copart Link	Current Bid	Year	Make	Model	Engine Type	Kilometers	Odometer
Link	$125 CAD	2009	FORD	F350 SUPER	6.4L 8	163559	ACTUAL
Link	$5,700 CAD	2011	OPEN	5TH WHEEL		0	NOT ACTUAL
Link	$0 CAD	2007	GRET	TRAILER		0	NOT ACTUAL
Link	$125 CAD	2003	HONDA	ODYSSEY EX	3.5L 6	0	EXEMPT
Link	$250 CAD	2008	INFINITI	EX35/JOURN	3.5L 6	150720	ACTUAL
Link	$125 CAD	2007	CHRYSLER	PT CRUISER	2.4L 4	187940	ACTUAL
Link	$125 CAD	2013	MINI	COOPER S	1.6L 4	36279	NOT ACTUAL

5. The second part of this project was then to scrape the local car listing website for the average price of the cars up for auction.
    a. Again, with this portion of the macro, I make the user enter from which city they would like pull the pricing data. Because John sells in both markets I made it a simple drop down menu for him to pick from. This user form also has a built in control so that an option has to be selected.



    b. Building the actual data retrieval part involved telling the macro to go to the website, search for the specific car, and then using an array to find and store all of the searched cars' prices. Using an array here allowed me to average out the prices of the cars. And I hyperlinked the search results to the average price.

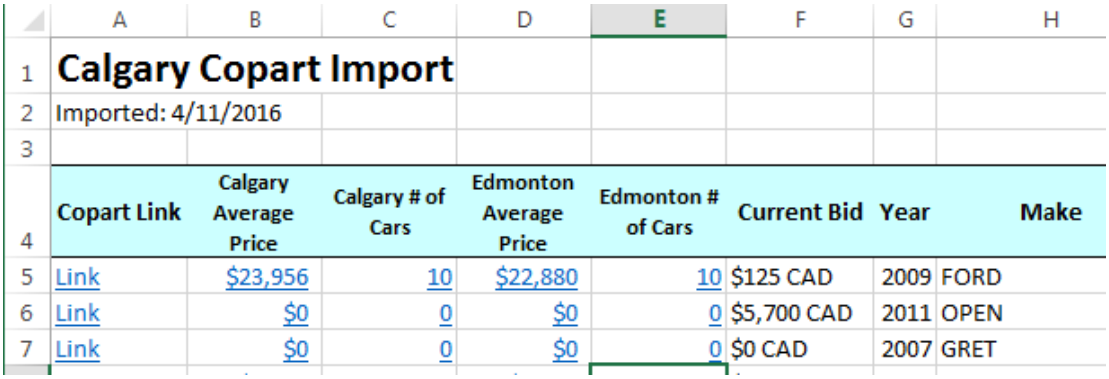


c. I also included how many cars were found in the search. This was something John wanted so that he could detect the popularity of the vehicle and know if the market was overly saturated.

d. I ended by formatting the two new columns that I inserted in the spreadsheet for the price and volume values.

1	**Calgary Copart Import**						
2	Imported: 4/11/2016						
3							
4	**Copart Link**	**Edmonton Average Price**	**Edmonton # of Cars**	**Current Bid**	**Year**	**Make**	**Model**
5	Link	$22,880	10	$125 CAD	2009	FORD	F350 SUPER
6	Link	$0	0	$5,700 CAD	2011	OPEN	5TH WHEEL
7	Link	$0	0	$0 CAD	2007	GRET	TRAILER

e. The other little feature I've added in the program is that the program will allow the user to run vehicle data from both cities and put the data in the same sheet. What I mean by this is that the program is programed in such a way that it allows for both cities to be searched and compared against just one auction.



	A	B	C	D	E	F	G	H
1	**Calgary Copart Import**							
2	Imported: 4/11/2016							
3								
4	**Copart Link**	**Calgary Average Price**	**Calgary # of Cars**	**Edmonton Average Price**	**Edmonton # of Cars**	**Current Bid**	**Year**	**Make**
5	Link	$23,956	10	$22,880	10	$125 CAD	2009	FORD
6	Link	$0	0	$0	0	$5,700 CAD	2011	OPEN
7	Link	$0	0	$0	0	$0 CAD	2007	GRET

Conceptual Difficulties

This project was very webpage heavy. And what I meant by that is that I learned how to navigate and pull data from websites. This included learning how to download a file off a website and telling the program where to put the downloaded file. It also heavily involved thinking through loops and how to pull website data. A big challenge I had was coming up with the appropriate website names for Kijiji. Kijiji for some reason has encoded their webpage url's. So each time I wanted to switch from Calgary to Edmonton I had to figure out what the code name was for that city and then write that in. At first it was a little confusing, but I figured it out. This was helpful because it did the same thing for the year and make of the cars. Each year and car needed certain numbers in the url in order for the webpage to execute.

Something that I didn't get to do in this project that I would have liked to done is to track on Kijiji how fast certain types of cars are selling. This of course would tell John which cars he could sell quickly, and help him focus his search on cars he could turn around quickly. This was something that in addition to this project would have been too lengthy, and quite frankly at the time he proposed it I didn't know how to go about doing it. But now after talking with Professor Allan and finishing the course, I think I could figure out how to make something like what John described.

Assistance

The only assistance I received was from Google, many vba forum boards, and Professor Allan. The problem that I could not figure out and had to go ask Professor Allan about was that Internet

Explorer was taking a long time to download and thus not allowing the program to go in and scrape the data. To fix this problem Professor Allan just changed the ready state to equal 3 or 4 instead of having it wait until 4 to go ahead and scrape. What was happening was that the webpages weren't getting to ready state 4, rather they were stopping at 3. This was why it seemed that my program was slowing down, because it was wasn't running at all! The webpages were still fine at ready state 3, we just had to change the code to recognize this change in state.