Python and Web Scraper

Emily Murphy (ekm4xz)

**Landing Data from Flightview.com**

**Introduction**

Through the use of the web scraper Beautiful Soup, I created a data frame with the status of flights arriving at O’Hare International Airport (ORD) in Chicago, IL. The data was scrapped from Flightview.com and included the airline, the flight number, arrival city, flight status, arrival terminal and gate, scheduled date/time of arrival, and updated date/time of arrival. Once the flight information was loaded into a CSV, I used the pandas library to read in the data and analyze the information. I created queries to find trends in airlines, arriving from city, and flight delays.

**Approach to Assignment**

My approach to the assignment was to find information on flights and scrap this into the file. I was hoping to perform some analysis to determine common flight paths and review different airlines based on the status of arriving flights, but this would be dependent on the information I was able to pull into Python.

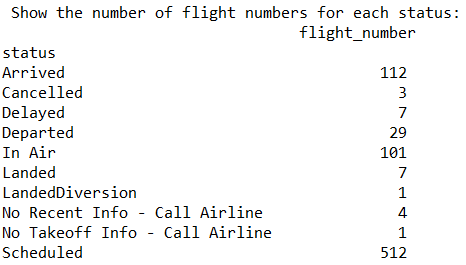
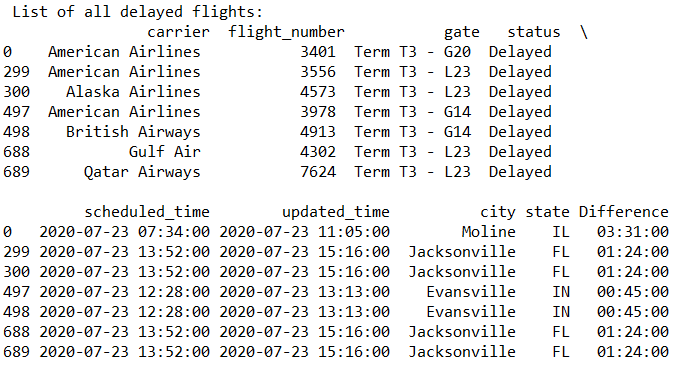
**Data Used**

The first step was to find a url with information I wanted to scrap. I chose flight status information from ORD because I have flown into ORD over a hundred times throughout the past four years of work. The website I chose included the airline, the flight number, arrival city, flight status, arrival terminal and gate, scheduled time of arrival, and updated time of arrival. To view the html of the table, I right clicked on the table and selected ‘From Source’. This html is where I found the url for the code because the html for the webpage did not include the specific row information. On an individual row, I right clicked and selected ‘Investigate’ to view the html code for the specific value in the table.

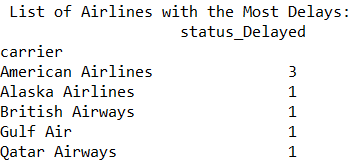
**Algorithms to Perform Analysis**

I utilized a video on Vimeo1 that walked through how to use BeautifulSoup. I worked to bring in the data for one value in one row at a time. Once I found the code that would give me the correct value from the html, I added this code to a ‘for’ loop. I repeated the process for each column in the first row, and then I used the ‘for’ loop to get the columns for every row of the data. I struggled to bring in the datetime information as it was a different format from the other text. I used an article from [www.crummy.com](http://www.crummy.com)2 to find other ways of bringing in html through BeautifulSoup and used the .next\_sibling method to move down the lines of html. I also used the .strptime code3 to put the information into the correct format, even though I later converted it back to a string to feed into the CSV. After I was able to print all the data, I wrote it to a CSV file using commas to separate the values in a column and “\n” to separate the rows. The from\_city field had commas that I converted to vertical bars based on what I learned in the Vimeo video because this would be read into separate columns by the CSV. Finally, I had to append data from a different class to the CSV4, but it used the same ‘for’ loop equations and headers.

Using pandas, I read the CSV file into a data frame, and I referenced <https://pandas.pydata.org/> for information throughout the analysis process. I used the functions .head(), .count(), and columns to ensure all the data appeared to have loaded correctly. I then wanted to change the vertical back to commas and split the data into two columns5, so it would be possible to do analysis by city or state. I created a list of all delayed flights and then created a table of the number of flight numbers per status.



I used the .nunique attribute6 to count the unique scheduled times for each city and used pd.get\_dummies on status to get binary values for each of the statuses. I used this to create a groupby and sum the number of delays each carrier has had.

**How to Use the Data**

There are many potential uses for my program for both airlines and travelers. One example is seeing how many flights from a specific city fly to ORD every day. I was surprised to see that Traverse City has as many flights to Chicago as Dallas-Fort Worth, TX. Chicago ORD and Dallas are both hubs for American Airlines, so I expect there to be many flights between the two cities. Traverse City is not a hub for any airline and is a relatively small city in Michigan, but it is a popular tourist destination in the Midwest. During the summer, more people would want to go on vacation there, so more frequent flights must have been scheduled. An airline could use this information to determine how many flights the other airlines are flying between two cities.

A traveler would appreciate being able to see the amount of flights at each status and which flights are delayed over 1 hour in arriving. The number of flights per each status shows the user how many flights have been canceled coming to ORD, which would impact the number of aircraft available for the flights out of ORD. As more flights arrive late, the departing flights using those planes will be delayed, so travelers should monitor the status of arriving flights when departing on a flight.

**Future Improvements**

My program could be improved by adding a unique ID column based on flight number and date and using this information to ensure there are no duplicates. The flight tracker is constantly updating the flight status, arrival date/time, and potentially even the gate of arrival. The online flight tracker also only shows flights for about a 6-hour time period, so the later flights appear on the website throughout the day. Currently, there is a risk of the same flight number and scheduled arrival time being listed twice if I run the program to append the updated data to the CSV.

Another update I would like to make is to determine a way to look at planes instead of flight numbers. There are multiple flight numbers for each flight, because the airline’s partners will have their own number for the flight. For example, one plane may be flying from Washington, DC and landing at Terminal 3 G7 on 7/23 at 8:40am. However, this flight could have 3 different flight numbers as it is operated as American Airlines but has code sharing agreements with British Airways and Alaska Airlines. While it is useful to include the individual flight numbers, this throws off the number of flights arriving. I was able to work around this for the count of flights from city using the count unique of scheduled time, because it is unlikely a flight from the same city will arrive at the exact same time at a different gate.

**Extra Credit**

I increased the functionality of the program by allowing the user to look up a flight and by calculating the difference between the scheduled arrival time and the updated arrival time. The user is able to enter the airline name and flight number separately. Based on this information, the flight’s details appear. If the user is waiting for a friend or family member to arrive, this allows the user to see if the flight is in flight or delayed. I also calculated the difference between the scheduled arrival time and the updated arrival time and created a new column called ‘Difference’7. This column allows the user to see what flights are delayed over an hour8. This column can also be summed for each airline to determine the total flight delays that have taken place. The top two airlines with the most delays are American Airlines and United Airlines, which makes sense given those airlines have the most flights to and from ORD.

**References**

1 https://vimeo.com/209499033

2 https://www.crummy.com/software/BeautifulSoup/bs4/doc/

3 https://stackoverflow.com/questions/43316608/remove-timestamp-from-date-string-in-python/43316706

4 https://stackoverflow.com/questions/2363731/append-new-row-to-old-csv-file-python

5 https://www.geeksforgeeks.org/split-a-text-column-into-two-columns-in-pandas-dataframe/?ref=rp

6 https://stackoverflow.com/questions/38309729/count-unique-values-with-pandas-per-groups

7 <https://stackoverflow.com/questions/37840812/pandas-subtracting-two-date-columns-and-the-result-being-an-integer>

8 <https://stackoverflow.com/questions/48376278/querying-timedelta-column-in-pandas-and-filtering-rows>