HW2 Part 2 - Flatting JSON data from Yelp

For this second part of the assignment we're going to be using the Yelp API via the python package `yelpapi`. Like with the Spotify API, you need to go get a yelp developer account here: https://www.yelp.com/fusion. When you try to create an app, it asks for name, description and contact info. The it gives you clientid and api key.

The goal will be to make a dataset of local taco places that are doing well during Coronavirus. Specifically, we want to make a dataset that takes their average score since they've been open and compares it to the average score of the last three reviews. Shops that have been doing well should hopefully have these two averages be similar.

There will be three main steps to this process:

- First you'll search by location type and get aggregate information for everything that falls in the ice cream category.
- Second you'll get reviews for all those locations. Yelp only returns three reviews when you call an ID, but that still works. Given you can only query one ID at a time, you'll need to write a loop to create a dataframe of all the reviews.
- Third, you'll aggregate the latest review information to see how their average review score compares to their overall average review score.

```
[48] # Mount google drive
     import os, sys
     from google.colab import drive
     drive.mount('/content/mnt')
     nb_path = '/content/notebooks'
     os.symlink('/content/mnt/My Drive/Colab Notebooks', nb_path)
     sys.path.insert(0, nb_path)  # or append(nb_path)
```

```
Mounted at /content/mnt
```

```
[ ] # Install yelpapi once. Next time you run this notebook, you can skip this.
    !pip install --target=$nb_path yelpapi
```

```
[49] !pip install yelpapi
```

```
Requirement already satisfied: yelpapi in /usr/local/lib/python3.10/dist-packages (2.5.1)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from yelpapi) (2.32.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->yelpapi) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->yelpapi) (3.8)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->yelpapi) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->yelpapi) (2024.8.30)
```

```
[50] # Now enter your API key so you can make requests
     from yelpapi import YelpAPI

     # yelp_api = YelpAPI('ENTER YOUR KEY HERE')
     api_key = '76E6F89aosfZor1yCwCVVD73TbLXxxqSDSB15TsUxU5GGMaXnxzsvhQ6qEUYTFkJTxRjcBHrRZY0MEad-SFEG6Cf5Fl4QYDmujlxny6ZQT2FGukqp8PHqvoVC_s7YXYx'
     yelp_api = YelpAPI(api_key)
```

Making calls to with Yelp API

There are many functions in the `yelpapi` package. The first one we'll use is `search_query()`. You can put in a term you want to search for followed up by the location and it'll give you all the locations that match. For example, the following would search for all ice cream places here in Tucson.

```
ice = yelp_api.search_query(term = 'ice cream', location = 'Tucson, AZ')
```

This will give you a response of all the ice cream places in Tucson. It'll be in JSON form so will need flattening before being useful.

Q6 Getting all taco shops and flattening - [3 points]

Task: Start by making a dataframe that uses the `search_query()` function to search using the term 'taco'. Call this `taco_shops`. After that, flatten the json results to `taco_shops_df`.

```
[51] ## Q6 part 1 Your code starts here
     # search for taco shops and store to taco_shops
     taco_shops = yelp_api.search_query(term = 'taco', location = 'Tucson, AZ')
```

```
[52] # Look at taco_shops
     taco_shops
```

```
    'categories': [{'alias': 'mexican', 'title': 'Mexican'}],
    'rating': 4.2,
    'coordinates': {'latitude': 32.1831398010254,
     'longitude': -110.97688293457},
    'transactions': ['pickup', 'delivery'],
    'price': '$',
    'location': {'address1': '3501 S 12th Ave',
     'address2': '',
     'address3': '',
```

```
[53] # What keys are present?
     taco_shops.keys()
     ## Q6 part 1  Your code ends here - Any code outside of these start/end markers won't be graded
```

```
dict_keys(['businesses', 'total', 'region'])
```

Now use the json_normalize function to flatten `taco_shops`. Note that you need to select the key that contains the businesses when flattening.
But this one is easier than the examples from the homework in that you don't need to provide any other arguments.

Store the result as `taco_shops_df`

After that select only the columns 'id', 'alias', 'name', 'review_count', and 'rating'.

```
## Q6 part 2 Your code starts here
# Flatten to taco_shops_df
import pandas as pd
from pandas import json_normalize

taco_shops_df = json_normalize(taco_shops['businesses'])
taco_shops_df.head() # Check
```

	id	alias	name	image_url	is_closed	url	review_count	categories	rating	transactions	...	coordinates.latitude	coordinates.lo
0	o3woQWQ-0HxFftltlEeNdw	el-rustico-tucson	El Rustico	https://s3-media1.fl.yelpcdn.com/bphoto/r0iQjo...	False	https://www.yelp.com/biz/el-rustico-tucson?adj...	289	[{'alias': 'tacos', 'title': 'Tacos'}]	4.5	[]	...	32.249041	-11
1	nvwtUKh6rBb29rFbC9kq5w	taco-rico-tucson-2	Taco Rico	https://s3-media3.fl.yelpcdn.com/bphoto/f9mKwX...	False	https://www.yelp.com/biz/taco-rico-tucson-2?ad...	99	[{'alias': 'mexican', 'title': 'Mexican'}, {'a...	4.9	[]	...	32.337110	-11
2	sYNKHA_o1eYurKLCxMOQnA	la-frida-mexican-grill-and-seafood-tucson	La Frida Mexican Grill & Seafood	https://s3-media4.fl.yelpcdn.com/bphoto/Y0REcH...	False	https://www.yelp.com/biz/la-frida-mexican-gril...	305	[{'alias': 'mexican', 'title': 'Mexican'}, {'a...	4.7	[]	...	32.206109	-11
3	jmwasbZfgj3honf79qKsnA	street-taco-and-beer-co-tucson	Street-Taco and Beer Co.	https://s3-media4.fl.yelpcdn.com/bphoto/VOaRTn...	False	https://www.yelp.com/biz/street-taco-and-beer-...	976	[{'alias': 'mexican', 'title': 'Mexican'}, {'a...	4.3	[delivery]	...	32.222001	-11
		tacos-	Tacos	https://s3-		https://www.yelp.com/biz/tacos-		[{'alias': 'mexican'		[pickup			

```
[54] 5 rows × 24 columns
```

```
[55] # Select only necessary columns
     taco_shops_df = taco_shops_df[['id', 'alias', 'name', 'review_count', 'rating']]
     taco_shops_df.shape # Check shape.  Is it 20 x 5?
     ## Q6 part 2  Your code ends here - Any code outside of these start/end markers won't be graded
```

```
(20, 5)
```

Q7 Getting reviews for the taco shops - [6 points]

Now we're going to use the `reviews_query()` function to get the last three reviews for a given ID. The issue here is that you can only feed it one ID at a time. So, we'll have to write a loop that queries for each ID in `taco_shops_df` and builds out a dataframe of reviews.

Task: Write a for loop that does the following steps:

- First make an empty data frame outside of the loop called `taco_shops_reviews_df`
- Initalize your loop so that it runs the length of `taco_shops_df` you made earlier.
- For each i in loop, use the id from `taco_shops_df` to get reviews from yelp using the `reviews_query()` function in `yelp_api` and store it to an object called `reviews`.
- Flatten `reviews` to an objected called `reviews_df`
- Add `location_id` to `reviews_df` - I gave you the code to do this :)
- Append `reviews_df` to `taco_shops_reviews_df` such that it builds out that dataframe with each `reviews_df` dataframe that's generated each loop.
- After the loop is done select only the columns `'id', 'text', 'rating', 'time_created', 'location_id'`

```
[67] ## Q7 Your code starts here
     # Write our loop

     taco_shops_reviews_df = pd.DataFrame()

     for i in range(len(taco_shops_df)):
         shop_id = taco_shops_df.iloc[i]['id']
         reviews = yelp_api.reviews_query(shop_id)
         reviews_df = pd.json_normalize(reviews, 'reviews')
         reviews_df['location_id'] = taco_shops_df['id'][i]
         taco_shops_reviews_df = pd.concat([taco_shops_reviews_df, reviews_df], ignore_index=True)
```

```
[68] # Select columns 'id', 'text', 'rating', 'time_created', 'location_id'
     taco_shops_reviews_df = taco_shops_reviews_df[['id', 'text', 'rating', 'time_created', 'location_id']]
     ## Q7  Your code ends here - Any code outside of these start/end markers won't be graded
```

Q8 Aggregating your review data - [3 points]

Task: Now go and do a data aggregation to get the mean review score across the three reviews. Remember, we want this grouped by location_id. Call this dataframe `latest_reviews_agg`.

In your groupby you should set `as_index` to false to make joining on the ID values easier.

You should also rename the second column to `mean_rating` vs. leaving it as the dual level name.

```
[69] ## Q8 Your code starts here
     # Do your groupby to get mean rating.
     # Call it latest_reviews_agg.
     latest_reviews_agg = taco_shops_reviews_df.groupby('location_id', as_index=False)['rating'].mean()
```

```
[70] # Check it!
     latest_reviews_agg.head()
```

	location_id	rating
0	0ghzROkZWKWwQKDt1fkPAQ	4.333333
1	A-5IN85MwL9F8wJRsDna6g	5.000000
2	DA5Mecz4auAhTTlFWNh33Q	4.666667
3	DVBJRvnCpkqaYl6nHroaMg	5.000000
4	Jcm-2RZWlcF0E4V6kekBsQ	4.666667

Next steps: Generate code with latest_reviews_agg | View recommended plots | New interactive sheet

```
[79] # Rename so the two column names are 'location_id' and 'mean_rating'
     latest_reviews_agg = latest_reviews_agg.rename(columns={'rating': 'mean_rating'})
     ## Q8  Your code ends here - Any code outside of these start/end markers won't be graded
```

Q9 Join your two datasets and one last transform - [3 points]

Task: Now it's time to join `latest_reviews_agg` back to your `taco_shops_df` dataframe. You're going to want to join them on the location_id, but remember it is called just 'id' in the `taco_shops_df` dataframe. Call this joined dataset `taco_shops_comp`

After you make that dataset do one last transform. In this I want you to make a new column called 'still_good' where the value is 'yes' if the mean_rating is greater than or equal to the average rating since they opened, or 'no' if the rating has dropped. The idea here is that this could be a value that one would use to see if their average score of the latest reviews has improved or suffered during Covid.

```
[80] ## Q9 Your code starts here
     # Join and name taco_shops_comp

     taco_shops_comp = pd.merge(taco_shops_df, latest_reviews_agg, left_on='id', right_on='location_id', how='left')
```

When you are comparing the columns make sure to not comparing taco_shops_df and taco_shops_comp since their tuples may have different orders! Use both columns from taco_shops_comp

```
[81] # Make still_good column
     taco_shops_comp['still_good'] = taco_shops_comp.apply(lambda row: 'yes' if row['mean_rating'] >= row['rating'] else 'no',axis=1)
```

```
[82] # So, do any locations have a lower average in their last three reviews compared to their average overall score?
     locations_with_lower_recent_ratings = taco_shops_comp[taco_shops_comp['still_good'] == 'no']
     print(locations_with_lower_recent_ratings)
     ## Q9  Your code ends here - Any code outside of these start/end markers won't be graded
```

```
                        id                                  alias  \
1   nvwtUKh6rBb29rFbC9kq5w                    taco-rico-tucson-2
2   sYNKHA_o1eYurKLCxMOQnA  la-frida-mexican-grill-and-seafood-tucson
5   aZ-2J_3nvuT-XdryCjrthg          carniceria-wild-west-tucson-2
8   nEaTbGFlU7d9eLU2kl6KBw          taqueria-el-pueblito-tucson-2
13  MK0OMY_u9unl8xSqjPLtMw                seis-kitchen-tucson-3
14  DA5Mecz4auAhTTlFWNh33Q                  la-chaiteria-tucson
17  sk11RFNogbE0axZUIDBRHw                 los-mezquites-tucson
18  orLNj9bDRMxaz6Hgs9YvVg                  agave-house-tucson-2

                          name  review_count  rating  \
1                   Taco Rico            99     4.9
2   La Frida Mexican Grill & Seafood     305     4.7
5              Carniceria Wild West      84     4.7
8              Taqueria El Pueblito     395     4.2
13                 Seis Kitchen        1000     4.2
14                 La Chaiteria         164     4.7
17                Los Mezquites          27     4.7
18                  Agave House          97     4.4
```

```
                  location_id  mean_rating still_good
1   nvwtUKh6rBb29rFbC9kq5w     4.333333         no
2   sYNKHA_o1eYurKLCxMOQnA     4.666667         no
5   aZ-2J_3nvuT-XdryCjrthg     4.000000         no
8   nEaTbGFlU7d9eLU2kl6KBw     3.000000         no
13  MK0OMY_u9unl8xSqjPLtMw     3.666667         no
14  DA5Mecz4auAhTTlFWNh33Q     4.666667         no
17  sk11RFNogbE0axZUIDBRHw     4.666667         no
18  orLNj9bDRMxaz6Hgs9YvVg     3.000000         no
```

You should get 8 shops (this may change suddenly since data is live, but on July 12 2024 is still 8) including El Rustico and Seis Kitchen