rec05

October 1, 2020

1 CS 1656 – Introduction to Data Science

1.1 Instructor: Alexandros Labrinidis / Teaching Assistant: Evangelos Karageorgos

1.1.1 Additional credits: Xiaoting Li, Tahereh Arabghalizi, Zuha Agha, Anatoli Shein, Phuong Pham

1.2 ## Recitation : SQL via Data API

In this recitation, you will execute SQL queries on real data by connecting to the open data portal of Western Pennsylavnia Regional Data Center and requesting data via API calls.

```
In [1]: import json
        from datetime import datetime, timedelta, date
        import requests
        import pandas as pd
        import matplotlib.pyplot as plt

        %matplotlib inline
```

We will be using Allegheny County Restaurant/Food Facility Inspection Violation Dataset found here https://data.wprdc.org/dataset/allegheny-county-restaurant-food-facility-inspection-violations. This dataset contains violation data from actual routine inspections by one of health department staff's members for the last two years. It should be fun to find out inspection results for places where we eat in Pittsburgh! =)

```
In [2]: wprdc_api_endpoint = "https://data.wprdc.org/api/3/action/datastore_search_sql"

        # id for database table
        resource_id = "1a1329e2-418c-4bd3-af2c-cc334e7559af"

        # Get the date from 270 days ago)
        # end_date = datetime.now()
        # start_date = end_date - timedelta(days=270)

        # Get two date endpoints
        start_date = date(2018, 9, 1)
        end_date = date(2019, 6, 1)
```

```
        # Convert to a string the format the the data center accepts (yyyy-mm-dd)
        start_str = start_date.strftime("%Y-%m-%d")
        end_str = end_date.strftime("%Y-%m-%d")

        # SQL query we'll use in API call to request data
        query = """
        SELECT *
        FROM "{}"
        WHERE "inspect_dt" BETWEEN '{}' and '{}' AND "city" = '{}'""".format(resource_id, start_

        # Make WPRDC API Call
        response = requests.get(wprdc_api_endpoint, {'sql': query})

        # Parse response JSON into python dictionary
        response_data = json.loads(response.text)

        # Convert dictionary to dataframe
        df = pd.DataFrame.from_dict(response_data['result']['records'])

        # Print the number of rows
        print(df.shape[0], "rows total")
        print(df.columns)
        df.head()
```

```
19245 rows total
Index(['_full_text', '_geom', '_id', '_the_geom_webmercator', 'bus_st_date',
       'city', 'description', 'description_new', 'encounter', 'end_time',
       'facility_name', 'high', 'id', 'inspect_dt', 'low', 'medium',
       'municipal', 'num', 'placard_st', 'rating', 'start_time', 'state',
       'street', 'url', 'zip'],
      dtype='object')
```

Out[2]:

	_full_text	_geom	_id	\
0	'-04':36 '-09':35 '-10':5 '-13':6 '/reports/rw...	None	1750681	
1	'-04':33 '-09':32 '-10':5 '-13':6 '/reports/rw...	None	1750682	
2	'-04':29 '-09':28 '-10':3 '-12':4 '-120':32 '/...	None	1750684	
3	'-04':33 '-09':32 '-10':5 '-12':6 '-120':36 '/...	None	1750685	
4	'-04':24 '-09':23 '-10':3 '-12':4 '-120':31 '/...	None	1750686	

	_the_geom_webmercator	bus_st_date	city	\
0	None	2010-10-13	Pittsburgh	
1	None	2010-10-13	Pittsburgh	
2	None	2015-10-12	Pittsburgh	
3	None	2015-10-12	Pittsburgh	
4	None	2015-10-12	Pittsburgh	

	description	\