Matthew Peetz

MSDS 621

Regis University

Week 6 Lab: APIs

Introduction

I have been lucky enough to live in Denver, CO my entire life. I am very interested to see how the climate has changed in recent years, specifically in terms of local rainfall. I was able to find precipitation data from a wheather station near my home from 2013 - 2023. I will be pulling that information off the NOAA site using their API and storing it in an excel file.

In [1]:
```
### Remove or comment out this cell
import pandas as pd

# loading my specific credentials
data = pd.read_csv('data_wk6/mipauth2.csv',header=0)

# setting up some variables for the API.
my_token = data['token'][0]
my_token
```

Out[1]:
```
'KuhdaKeCidqAAzJqutEGIYqgucDPTzga'
```

Below is a picture of the weather station and its location.

View the Results of Your Query

To see the results of your specific query, click on the station icon. From the Find A Station results, we will need to capture the following details:

- Capture the values within the 'Network' and 'Id' fields (second cell from top, split on ':')

Daily Summaries Station Details

STATION DETAILS	
Name	LAKEWOOD 2.2 ESE, CO US
Network:ID	GHCND:US1COJF0179
Latitude/Longitude	39.68353°, -105.07579°
Elevation	1681 m

In [2]:
```
# variables based on my station search
network = 'GHCND'
ID = 'US1COAR0246'

# station_id = network:ID
station_id = network + ':' + ID
print(station_id)
```

```
GHCND:US1COAR0246
```

What type of data are we looking for?

The data set includes a couple of interesting features:

- PRCP - total liquid precipitation for the year
- WESD - water equivalent of snow on the ground
- WESF - water equivalent of snowfall

Testing to see if there is a connection to the sites API and what the dictionary key values are

In [3]:
```
import requests
import json

# building the parameter dictionary
# 'limit = 1000' --> What does this do? Look at the NOAA API documentation
data = {}
data = {'limit':'1000', 'datasetid': network, 'station_id': station_id}

# calling NOAA API to get the available datatypes for this specific static
r = requests.get('https://www.ncdc.noaa.gov/cdo-web/api/v2/datatypes',para
```

Now we need to convert the JSON output from the request to something more readable

In [4]:
```
# JSON to dictionary
datatypes_dict = json.loads(r.text)

# need the keys from this dictionary
datatypes_dict.keys()

```

Out[4]:
```
dict_keys(['metadata', 'results'])
```

In [5]:
```
1 datatypes_dict['results'][:10]
```

Out[5]:
```
[{'mindate': '1994-03-19',
  'maxdate': '1996-05-28',
  'name': 'Average cloudiness midnight to midnight from 30-second ceilometer
data',
  'datacoverage': 1,
  'id': 'ACMC'},
 {'mindate': '1965-01-01',
  'maxdate': '2005-12-31',
  'name': 'Average cloudiness midnight to midnight from manual observations',
  'datacoverage': 1,
  'id': 'ACMH'},
 {'mindate': '1994-02-01',
  'maxdate': '1996-05-28',
  'name': 'Average cloudiness sunrise to sunset from 30-second ceilometer dat
a',
  'datacoverage': 1,
  'id': 'ACSC'},
 {'mindate': '1965-01-01',
  'maxdate': '2005-12-31',
  'name': 'Average cloudiness sunrise to sunset from manual observations',
  'datacoverage': 1,
  'id': 'ACSH'},
 {'mindate': '1982-01-01',
  'maxdate': '2023-07-23',
  'name': 'Average wind speed',
  'datacoverage': 1,
  'id': 'AWND'},
 {'mindate': '1948-08-02',
  'maxdate': '2012-07-23',
  'name': 'Number of days included in the multiday evaporation total (MDEV)',
  'datacoverage': 1,
  'id': 'DAEV'},
 {'mindate': '1832-05-11',
  'maxdate': '2023-07-23',
  'name': 'Number of days included in the multiday precipitation total (MDP
R)',
  'datacoverage': 1,
  'id': 'DAPR'},
 {'mindate': '1877-01-02',
  'maxdate': '2022-11-20',
  'name': 'Number of days included in the multiday snow fall total (MDSF) ',
  'datacoverage': 1,
  'id': 'DASF'},
 {'mindate': '1863-05-04',
  'maxdate': '2023-07-15',
  'name': 'Number of days included in the multiday minimum temperature (MDT
N)',
  'datacoverage': 1,
  'id': 'DATN'},
 {'mindate': '1863-05-04',
  'maxdate': '2023-07-14',
  'name': 'Number of days included in the multiday maximum temperature (MDT
X)',
  'datacoverage': 1,
```

```
 'id': 'DATX'}]
```

Setting up the data requests for the API

In [6]:
```
data = {}
data = {'limit':'1000', 'datasetid': network, 'stationid': station_id}


# append additional parameters to data dictionary
data.update({'datatypeid': 'PRCP, WESD, WESF'})
data.update({'startdate': '2013-05-14'})
data.update({'enddate': '2013-12-31'})
data.update({'units':'standard'})
data
```

Out[6]:
```
{'limit': '1000',
 'datasetid': 'GHCND',
 'stationid': 'GHCND:US1COAR0246',
 'datatypeid': 'PRCP, WESD, WESF',
 'startdate': '2013-05-14',
 'enddate': '2013-12-31',
 'units': 'standard'}
```

Pulling the results for 2013 from the website

In [7]:
```
# make the request to get our year of data
r = requests.get('https://www.ncdc.noaa.gov/cdo-web/api/v2/data',params =

#load the api response as a json
prcp_2013_dict = json.loads(r.text)
```

In [8]:
```
# look at the first record of our data
prcp_2013_dict['results'][:5]
```

Out[8]:
```
[{'date': '2013-05-14T00:00:00',
  'datatype': 'PRCP',
  'station': 'GHCND:US1COAR0246',
  'attributes': ',,N,0730',
  'value': 0.0},
 {'date': '2013-05-15T00:00:00',
  'datatype': 'PRCP',
  'station': 'GHCND:US1COAR0246',
  'attributes': 'T,,N,0700',
  'value': 0.0},
 {'date': '2013-05-16T00:00:00',
  'datatype': 'PRCP',
  'station': 'GHCND:US1COAR0246',
  'attributes': ',,N,0730',
  'value': 0.04},
 {'date': '2013-05-17T00:00:00',
  'datatype': 'PRCP',
  'station': 'GHCND:US1COAR0246',
  'attributes': ',,N,0700',
  'value': 0.0},
 {'date': '2013-05-20T00:00:00',
  'datatype': 'PRCP',
  'station': 'GHCND:US1COAR0246',
  'attributes': ',,N,0800',
  'value': 0.04}]
```

Putting the dictionary into a pandas data frame

In [9]:
```
df = pd.DataFrame.from_dict(prcp_2013_dict['results'])
df
```

Out[9]:

	date	datatype	station	attributes	value
0	2013-05-14T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.00
1	2013-05-15T00:00:00	PRCP	GHCND:US1COAR0246	T,,N,0700	0.00
2	2013-05-16T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.04
3	2013-05-17T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0700	0.00
4	2013-05-20T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0800	0.04
...	...	...	...	...	...
211	2013-12-18T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.00
212	2013-12-19T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0830	0.00
213	2013-12-26T00:00:00	WESD	GHCND:US1COAR0246	,,N,0900	0.00
214	2013-12-27T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0900	0.00
215	2013-12-28T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0900	0.00

216 rows × 5 columns

Writing the 2013 data into a csv file

In [10]:
```
# Writing DataFrame to a CSV file
df.to_csv("output.csv", index=False)
```

In [11]:
```
df
```

Out[11]:

	date	datatype	station	attributes	value
0	2013-05-14T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.00
1	2013-05-15T00:00:00	PRCP	GHCND:US1COAR0246	T,,N,0700	0.00
2	2013-05-16T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.04
3	2013-05-17T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0700	0.00
4	2013-05-20T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0800	0.04
...	...	...	...	...	...
211	2013-12-18T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.00
212	2013-12-19T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0830	0.00
213	2013-12-26T00:00:00	WESD	GHCND:US1COAR0246	,,N,0900	0.00
214	2013-12-27T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0900	0.00
215	2013-12-28T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0900	0.00

216 rows × 5 columns

Pulling the remaining data

The API will only let you pull one year of data, so I will need to set up a loop to pull all the data, add it to the dictionary, and then write it to the csv file

In [12]:

```
# Loop to get all years data

year = 2014

network = 'GHCND'
ID = 'US1COAR0246'

# station_id = network:ID
station_id = network + ':' + ID

# creating empty set for holding data

while year < 2024:
    data = {}
    data = {'limit':'1000', 'datasetid': network, 'stationid': station_id}
    data.update({'datatypeid': 'PRCP, WESD, WESF'})
    data.update({'startdate': str(year) + '-05-14'})
    data.update({'enddate': str(year) + '-12-31'})
    data.update({'units':'standard'})
    data

    r = requests.get('https://www.ncdc.noaa.gov/cdo-web/api/v2/data',param

    # load into json file
    info = json.loads(r.text)
    print(info['results'][:1])
    df_temp = pd.DataFrame.from_dict(info['results'])
    df = pd.concat([df, df_temp])


    #print(data)
    year = int(year + 1)
    #print(year)
```

```
[{'date': '2014-05-22T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0800', 'value': 0.15}]
[{'date': '2015-05-24T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0800', 'value': 0.57}]
[{'date': '2016-05-26T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0830', 'value': 1.01}]
[{'date': '2017-05-14T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0700', 'value': 0.0}]
[{'date': '2018-06-19T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0700', 'value': 0.0}]
[{'date': '2019-06-23T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,1133', 'value': 0.95}]
[{'date': '2020-05-15T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': 'T,,N,0700', 'value': 0.0}]
[{'date': '2021-06-20T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': 'T,,N,0700', 'value': 0.0}]
[{'date': '2022-07-26T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,1720', 'value': 3.0}]
[{'date': '2023-07-02T00:00:00', 'datatype': 'PRCP', 'station': 'GHCND:US1COA
R0246', 'attributes': ',,N,0700', 'value': 0.0}]
```

Taking a look at the data frame

In [13]:
```
1 df
```

Out[13]:

	date	datatype	station	attributes	value
0	2013-05-14T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.00
1	2013-05-15T00:00:00	PRCP	GHCND:US1COAR0246	T,,N,0700	0.00
2	2013-05-16T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0730	0.04
3	2013-05-17T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0700	0.00
4	2013-05-20T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0800	0.04
...	...	...	...	...	...
3	2022-10-01T00:00:00	PRCP	GHCND:US1COAR0246	,,N,1146	1.22
4	2022-10-02T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0700	0.18
5	2022-10-03T00:00:00	PRCP	GHCND:US1COAR0246	T,,N,0700	0.00
0	2023-07-02T00:00:00	PRCP	GHCND:US1COAR0246	,,N,0700	0.00
1	2023-07-15T00:00:00	PRCP	GHCND:US1COAR0246	T,,N,0700	0.00

375 rows × 5 columns

In [14]:
```
1 # Writing DataFrame to a CSV file
2 df.to_csv("output2.csv", index=False)
```

I'm going to guess that the information we are after is stored in the results key. Let's look at the first 5 and see if we might be right

Conclusion

An API was created to access weather data from a single station on the NOAA website. Information was then pulled using that weather station location and data type interested in, precipitation and snow fall. That data was then loaded into a pandas dataframe and put into a csv file for further analysis in the future.

In [ ]:
```
1
```