Collecting weather data from an API

About the data

In this notebook, we will be collecting daily weather data from the National Centers for Environmental Information (NCEI) API. We will use the Global Historical Climatology Network - Daily (GHCND) data set; see the documentation here.

Note: The NCEI is part of the National Oceanic and Atmospheric Administration (NOAA) and, as you can see from the URL for the API, this resource was created when the NCEI was called the NCDC. Should the URL for this resource change in the future, you can search for the NCEI weather API to find the updated one.

Using the NCEI API

Paste your token below.

```
import requests

def make_req(endpoint, payload=None):
  return requests.get(
      f'https://www.ncdc.noaa.gov/cdo-web/api/v2/{endpoint}',
      headers={
          'token' : 'COGqDwKIRDvdlxMHVhxgqTsoJqtCoAfz'
      },params=payload
  )
```

```
response = make_req('datasets', {'startdate':'2018-10-01'})
response.status_code

# check if API is working
```

```
200
```

Collect All Data Points for 2018 In NYC (Various Stations)

We can make a loop to query for all the data points one day at a time. Here we create a list of all the results:

```
import datetime

from IPython import display as dis

current = datetime.date(2018,1,1)
end = datetime.date(2019,1,1)

results = []

while current < end :
  dis.clear_output(wait=True)
  dis.display(f'gathering data for {str(current)}')

  response = make_req(
      'data',{
          'datasetid':'GHCND',
          'locationid':'CITY:US360019',
          'startdate':current,
          'enddate':current,
          'units':'metric',
          'limit':1000
      }
  )
  if response.ok:
    results.extend(response.json()['results'])

  current += datetime.timedelta(days=1)
```

```
'gathering data for 2018-12-31'
```

Now, we can create a dataframe with all this data. Notice there are multiple stations with values for each datatype on a given day. We don't know what the stations are, but we can look them up and add them to the data:

```
import pandas as p

df=p.DataFrame(results)
df
```

	date	datatype	station	attributes	value
0	2018-01-01T00:00:00	PRCP	GHCND:US1CTFR0039	,,N,0800	0.0
1	2018-01-01T00:00:00	PRCP	GHCND:US1NJBG0015	,,N,1050	0.0
2	2018-01-01T00:00:00	SNOW	GHCND:US1NJBG0015	,,N,1050	0.0
3	2018-01-01T00:00:00	PRCP	GHCND:US1NJBG0017	,,N,0920	0.0
4	2018-01-01T00:00:00	SNOW	GHCND:US1NJBG0017	,,N,0920	0.0
...	...	...	...	...	...
91317	2018-12-31T00:00:00	WDF5	GHCND:USW00094789	,,W,	130.0
91318	2018-12-31T00:00:00	WSF2	GHCND:USW00094789	,,W,	9.8
91319	2018-12-31T00:00:00	WSF5	GHCND:USW00094789	,,W,	12.5
91320	2018-12-31T00:00:00	WT01	GHCND:USW00094789	,,W,	1.0
91321	2018-12-31T00:00:00	WT02	GHCND:USW00094789	,,W,	1.0

91322 rows × 5 columns

Next steps: View recommended plots

Save this data to a file:

```
df.to_csv('/content/sample_data/nycweather2k18.csv', index=False)
```

and write it to the database:

```
import sqlite3 as sq3

with sq3.connect('/content/weather.db') as connection:
  df.to_sql(
      'weather',connection, index=False, if_exists='replace'
  )
```

For learning about merging dataframes, we will also get the data mapping station IDs to information about the station:

```
response = make_req(
    'stations',{
        'datasetid':'GHCND',
        'locationid':'CITY:US360019',
        'limit':1000
    }
)
# serached using the database(SQLite3) connection from previous cell ↑↑↑
stations = p.DataFrame(response.json()['results'])[['id','name','latitude','longitude','elevation']]
stations.to_csv('/content/sample_data/weather_stations.csv', index=False) # create a csv of the weather stations

with sq3.connect('/content/weather.db') as connection:
  stations.to_sql('stations',connection,index=False, if_exists='replace')
```

```
ws = p.read_csv('/content/sample_data/weather_stations.csv')
ws # weather stations
```

	id	name	latitude	longitude	elevation
0	GHCND:US1CTFR0022	STAMFORD 2.6 SSW, CT US	41.064100	-73.577000	36.6
1	GHCND:US1CTFR0039	STAMFORD 4.2 S, CT US	41.037788	-73.568176	6.4
2	GHCND:US1NJBG0001	BERGENFIELD 0.3 SW, NJ US	40.921298	-74.001983	20.1
3	GHCND:US1NJBG0002	SADDLE BROOK TWP 0.6 E, NJ US	40.902694	-74.083358	16.8
4	GHCND:US1NJBG0003	TENAFLY 1.3 W, NJ US	40.914670	-73.977500	21.6
...	...	...	...	...	...
315	GHCND:USW00054787	FARMINGDALE REPUBLIC AIRPORT, NY US	40.734430	-73.416370	22.8
316	GHCND:USW00094728	NY CITY CENTRAL PARK, NY US	40.778980	-73.969250	42.7
317	GHCND:USW00094741	TETERBORO AIRPORT, NJ US	40.858980	-74.056160	0.8
318	GHCND:USW00094745	WESTCHESTER CO AIRPORT, NY US	41.062360	-73.704540	112.9
319	GHCND:USW00094789	JFK INTERNATIONAL AIRPORT, NY US	40.639150	-73.763900	2.7

320 rows × 5 columns

Next steps: View recommended plots