Collecting weather data from an API

Using the NCEI API

```
import requests

def make_request(endpoint, payload=None):
  """
  Make a request to a specific endpoint on the weather API
  passing headers and optional payload.

  Parameters:
    - endpoint: The endpoint of the API you want to
               make a GET request to.
    - payload: A dictionary of data to pass along
              with the request.
  Returns:
    Response object.
  """
  return requests.get(
      f'https://www.ncdc.noaa.gov/cdo-web/api/v2/{endpoint}',
      headers={
          'token': 'UNIsVJdJBXcATKDpvertADsBLGtQbFwP'
      },
      params=payload
  )
```

Disk: 24.40 GB/107.72 GB

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

```
# Demonstrates a data gathering process where it will gather from the NCEI API
# weather data from the various NYC stations starting from the year 2018 to 2019.
import datetime

from IPython import display # for updating the cell dynamically

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

results = []

while current < end:
  #update the cell with status information
  display.clear_output(wait=True)
  display.display(f'gathering data for {str(current)}')

  response = make_request(
      'data',
      {
          'datasetid' : 'GHCND', # Global Historical Climatology Network - Daily (GHCND) dataset
          'locationid' : 'CITY:US360019', # NYC
          'startdate' : current,
          'enddate' : current,
          'units' : 'metric',
          'limit' : 1000 # max allowed

      }
  )

  if response.ok:
      # we extend the l Disk: 24.40 GB/107.72 GB avoid getting a nested list
      results.extend(response.json()['results'])

  # update the current date to avoid an infinite loop
  current += datetime.timedelta(days=1)
```

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

```
# The gathered data from the previous process will be converted into a dataframe
# named df.
import pandas as pd
df = pd.DataFrame(results)
df.head()
```

	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

Next steps: View recommended plots

```
# will save the nyc_weather_2018 csv file to the sample_data folder
df.to_csv('/content/sample_data/nyc_weather_2018.csv', index=False)
```

```
# converts the df dataframe to a SQLite database file
# where it will be saved to the sample_data folder
import sqlite3
with sqlite3.connect('/content/sample_data/weather.db') as connection:
  df.to_sql(
    'weather', connection, index=False, if_exists='replace'
  )
```

```
# will collect weather data from 1000 NYC stations that came from the NCEI API
# the json file will be converted into csv that will be converted again into a
# SQLite db and will replace the previous weather.db file
response = make_request(
  'stations',
  {
    'datasetid' : 'GHCND', # Global Historical Climatology Network - Daily (GHCND) dataset
    'locationid' : 'CITY:US360019', # NYC
    'limit' : 1000 # max allowed
  }
)
stations = pd.DataFrame(response.json()['results'])[['id', 'name', 'latitude', 'longitude', 'elevation']]
```

Disk: 24.40 GB/107.72 GB