Automated Weather Pull / Pandas Introduction

Joshua Skipper

In [27]:

```
import requests

#Create an account @ https://openweathermap.org/
# Grab the API Key from your account
API_KEY = 'd1a650f175b0d849b7cc772b25ff66a9'
BASE_URL = "http://api.openweathermap.org/data/2.5/weather"

city = input("Enter a city name: \n")
request_url = f"{BASE_URL}?appid={API_KEY}&q={city}"
response = requests.get(request_url)

print(response.json())
```

```
Enter a city name: 
Atlanta
{'coord': {'lon': -84.388, 'lat': 33.749}, 'weather': [{'id': 500, 'main': 'Rain', 'description': 'light rain', 'icon': '10d'}], 'base': 'stations', 'main': {'temp': 290.22, 'feels_like': 290.05, 'temp_min': 287.98, 'temp_m
ax': 292.13, 'pressure': 1015, 'humidity': 79}, 'visibility': 10000, 'wind': {'speed': 4.63, 'deg': 100}, 'rain': {'1h': 0.92}, 'clouds': {'all': 100}, 'dt': 1681402029, 'sys': {'type': 2, 'id': 2006620, 'country': 'US', 's
unrise': 1681384168, 'sunset': 1681430779}, 'timezone': -14400, 'id': 4180439, 'name': 'Atlanta', 'cod': 200}
```

In [28]:

```
# Clean / Convert your data into readable text
if response.status_code == 200:
    data = response.json()
    weather = data['weather'][0]['description']
    celcius = round(data['main']['temp'] - 273.15)
    fahrenheit = round((celcius * 9/5) + 32)
    print(f"This is weather report for {city}.")
    print(f"Weather: {weather}")
    print(f"Temperature: {fahrenheit} in fahrenheit")
    print(f"Temperature: {celcius} in celcius")
else:
    print("An error occurred.")
```

```
This is weather report for Atlanta.
Weather: light rain
Temperature: 63 in fahrenheit
Temperature: 17 in celcius
```

In [20]:

```
import pandas as pd

#Putting the data into a dataframe
df = pd.json_normalize(data['main'])

#Add new column for a current timestamp
df['timestamp'] = pd.to_datetime('now')
df
```

```
C:\Users\skipper001\AppData\Local\Temp\ipykernel_18760\3662476903.py:7: FutureWarning: The parsing of 'now' in pd.to_datetime without `utc=True` is deprecated. In a future version, this will match Timestamp('now') and Times
tamp.now()
  df['timestamp'] = pd.to_datetime('now')
```

Out[20]:

	temp	feels_like	temp_min	temp_max	pressure	humidity	timestamp
0	290.22	290.05	287.98	292.13	1015	79	2023-04-13 16:10:26.577248

In [29]:

```
city = input("Enter a city name: \n")
request_url = f"{BASE_URL}?appid={API_KEY}&q={city}"
response = requests.get(request_url)

print(response.json())
```

```
Enter a city name: 
New York
{'coord': {'lon': -74.006, 'lat': 40.7143}, 'weather': [{'id': 800, 'main': 'Clear', 'description': 'clear sky', 'icon': '01d'}], 'base': 'stations', 'main': {'temp': 301.68, 'feels_like': 300.5, 'temp_min': 298.79, 'temp_m
ax': 304.16, 'pressure': 1013, 'humidity': 27}, 'visibility': 10000, 'wind': {'speed': 6.17, 'deg': 290, 'gust': 8.75}, 'clouds': {'all': 0}, 'dt': 1681401959, 'sys': {'type': 2, 'id': 2008101, 'country': 'US', 'sunrise': 1
681381231, 'sunset': 1681428734}, 'timezone': -14400, 'id': 5128581, 'name': 'New York', 'cod': 200}
```

In [30]:

```
if response.status_code == 200:
    data = response.json()
    weather = data['weather'][0]['description']
    celcius = round(data['main']['temp'] - 273.15)
    fahrenheit = round((celcius * 9/5) + 32)
    print(f"This is weather report for {city}.")
    print(f"Weather: {weather}")
    print(f"Temperature: {fahrenheit} in fahrenheit")
    print(f"Temperature: {celcius} in celcius")
else:
    print("An error occurred.")
```

```
This is weather report for New York.
Weather: clear sky
Temperature: 84 in fahrenheit
Temperature: 29 in celcius
```

In [ ]: