NASA_power_weather_data

September 12, 2024

Creating Karnataka weather datasets from NASA power project

by

Ramesh Kestur, IIIT-B

Data source

NASA power project

This is an implementation to use the APIs to collect weather data for all the 31 districts of Karnataka

0.1 Get all the parameters and check for Bangalore district

[7]:
```
import requests
import pandas as pd
from io import StringIO

# Define the district and its coordinates
district = "Bangalore"
coords = {"lat": 12.9716, "lon": 77.5946}

# Date range
start_date = "20200101"
end_date = "20231231"

# List of parameters to check
parameters = ["T2M", "WS2M", "PRECTOTCORR", "RH2M", "ALLSKY_SFC_SW_DWN",␣
 ↪"T2M_MIN", "T2M_MAX", "PS"]

# NASA POWER API URL template
url_template = "https://power.larc.nasa.gov/api/temporal/daily/point?
 ↪parameters={param}&community=AG&longitude={lon}&latitude={lat}&start={start}&end={end}&form
```

```
# Loop through each parameter and check for errors
for param in parameters:
    # Construct the URL with the current parameter
    url = url_template.format(param=param, lon=coords["lon"],␣
 ↪lat=coords["lat"], start=start_date, end=end_date)

    # Send the API request
    response = requests.get(url)

    # Check the response status code
    if response.status_code == 200:
        try:
            # Load the response text into a pandas DataFrame (skip 11 header␣
 ↪lines)
            data = pd.read_csv(StringIO(response.text), skiprows=11)
            print(f"Parameter '{param}' is valid. Data fetched successfully.")
        except pd.errors.ParserError:
            print(f"Parameter '{param}' returned an error while parsing CSV.")
    else:
        print(f"Parameter '{param}' is invalid or returned an error. Status␣
 ↪code: {response.status_code}")
```

```
Parameter 'T2M' is valid. Data fetched successfully.
Parameter 'WS2M' is valid. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid. Data fetched successfully.
Parameter 'RH2M' is valid. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid. Data fetched successfully.
Parameter 'T2M_MIN' is valid. Data fetched successfully.
Parameter 'T2M_MAX' is valid. Data fetched successfully.
Parameter 'PS' is valid. Data fetched successfully.
```

[ ]:

0.2 Print the raw API response format

[8]:
```
import requests

# Define the district and its coordinates
district = "Bangalore"
coords = {"lat": 12.9716, "lon": 77.5946}

# Date range
start_date = "20200101"
end_date = "20201231"

# List of parameters to check
```

```
parameters = ["T2M", "WS2M", "PRECTOTCORR", "RH2M", "ALLSKY_SFC_SW_DWN",␣
 ↪"T2M_MIN", "T2M_MAX", "PS"]

# NASA POWER API URL template
url_template = "https://power.larc.nasa.gov/api/temporal/daily/point?
 ↪parameters={param}&community=AG&longitude={lon}&latitude={lat}&start={start}&end={end}&form

# Fetch data for one parameter (e.g., T2M)
param = "T2M"
url = url_template.format(param=param, lon=coords["lon"], lat=coords["lat"],␣
 ↪start=start_date, end=end_date)

# Send the API request
response = requests.get(url)

# Print the first 10 lines of the response text (raw response including headers)
if response.status_code == 200:
    # Split response text by lines
    response_lines = response.text.splitlines()

    # Print the first 10 lines for debugging
    print(f"Raw API response for parameter '{param}':\n")
    for i, line in enumerate(response_lines[:50]):
        print(f"Line {i+1}: {line}")
else:
    print(f"Failed to fetch data. Status code: {response.status_code}")
```

```
Raw API response for parameter 'T2M':

Line 1: -BEGIN HEADER-
Line 2: NASA/POWER CERES/MERRA2 Native Resolution Daily Data
Line 3: Dates (month/day/year): 01/01/2020 through 12/31/2020
Line 4: Location: Latitude  12.9716   Longitude 77.5946
Line 5: Elevation from MERRA-2: Average for 0.5 x 0.625 degree lat/lon region =
841.72 meters
Line 6: The value for missing source data that cannot be computed or is outside
of the sources availability range: -999
Line 7: Parameter(s):
Line 8: T2M     MERRA-2 Temperature at 2 Meters (C)
Line 9: -END HEADER-
Line 10: YEAR,DOY,T2M
Line 11: 2020,1,22.9
Line 12: 2020,2,23.45
Line 13: 2020,3,23.88
Line 14: 2020,4,24.38
Line 15: 2020,5,24.15
Line 16: 2020,6,23.65
```

```
Line 17: 2020,7,22.47
Line 18: 2020,8,22.33
Line 19: 2020,9,22.58
Line 20: 2020,10,21.96
Line 21: 2020,11,21.19
Line 22: 2020,12,21.36
Line 23: 2020,13,21.97
Line 24: 2020,14,21.46
Line 25: 2020,15,20.44
Line 26: 2020,16,20.44
Line 27: 2020,17,21.51
Line 28: 2020,18,23.11
Line 29: 2020,19,23.48
Line 30: 2020,20,23.02
Line 31: 2020,21,21.87
Line 32: 2020,22,21.97
Line 33: 2020,23,22.6
Line 34: 2020,24,22.12
Line 35: 2020,25,21.42
Line 36: 2020,26,22.09
Line 37: 2020,27,22.83
Line 38: 2020,28,23.76
Line 39: 2020,29,23.81
Line 40: 2020,30,24.26
Line 41: 2020,31,24.11
Line 42: 2020,32,23.79
Line 43: 2020,33,23.64
Line 44: 2020,34,23.56
Line 45: 2020,35,23.62
Line 46: 2020,36,23.76
Line 47: 2020,37,24.05
Line 48: 2020,38,24.51
Line 49: 2020,39,25.4
Line 50: 2020,40,24.3
```

[ ]:

Detailed description of the fields from the NASA POWER API: 1. T2M (Temperature at 2 Meters)

```
Description: This represents the daily average air temperature measured at 2 meters above the g
Unit: Degrees Celsius (°C).
Use Case: This is one of the most common weather parameters and is essential for understanding
```

2. WS2M (Wind Speed at 2 Meters)

   Description: This is the average daily wind speed measured at 2 meters above the ground. Unit: Meters per second (m/s). Use Case: Wind speed data is critical for evaluating wind energy potential and understanding its influence on climate, transportation, and agriculture. Wind affects the rate of evaporation, and higher wind speeds can increase the cooling effect

on crops.

3. PRECTOTCORR (Precipitation Corrected)

   Description: This is the total daily precipitation, which includes all forms of liquid and solid water (rain, snow, hail) falling to the surface. The value is corrected to account for inconsistencies in precipitation measurement. Unit: Millimeters (mm) per day. Use Case: Precipitation is essential for water resource management, agriculture, flood risk assessment, and climate studies. Understanding how much water is available helps plan irrigation and anticipate flooding or drought conditions.

4. RH2M (Relative Humidity at 2 Meters)

   Description: This is the average daily relative humidity measured at 2 meters above the ground. Relative humidity is the amount of moisture in the air compared to the maximum amount of moisture the air can hold at that temperature. Unit: Percentage (%). Use Case: Relative humidity is important for comfort, agricultural planning, and understanding evaporation and plant transpiration rates. High humidity levels can increase the risk of fungal diseases in crops.

5. ALLSKY_SFC_SW_DWN (All Sky Surface Shortwave Downward Irradiance)

   Description: This measures the total solar radiation (shortwave) that reaches the Earth's surface under all sky conditions (including clear and cloudy skies). It represents the amount of sunlight that hits the ground. Unit: Kilowatt-hours per square meter per day ($kWh/m^2/day$). Use Case: Solar radiation data is essential for evaluating the potential for solar energy generation, as well as its effect on agriculture, ecosystems, and climate studies.

6. T2M_MIN (Minimum Temperature at 2 Meters)

   Description: This is the daily minimum air temperature measured at 2 meters above the ground. It reflects the coldest point in the day, typically during the early morning hours. Unit: Degrees Celsius (°C). Use Case: Minimum temperature data is important for evaluating frost risk in agriculture, especially for crops sensitive to cold conditions. It also helps in understanding daily temperature variability and trends.

7. T2M_MAX (Maximum Temperature at 2 Meters)

   Description: This is the daily maximum air temperature measured at 2 meters above the ground. It reflects the hottest point in the day, typically during the afternoon. Unit: Degrees Celsius (°C). Use Case: Maximum temperature data is critical for heat stress evaluation, especially in agriculture and public health. High temperatures can stress crops and livestock and can impact human comfort and energy consumption for cooling.

8. PS (Surface Pressure)

   Description: This is the daily surface pressure, which is the atmospheric pressure exerted by the weight of the atmosphere at the Earth's surface. Surface pressure is a key indicator of weather patterns. Unit: Pascals (Pa). Use Case: Surface pressure is used in weather forecasting and understanding large-scale atmospheric circulation patterns. Changes in surface pressure are often associated with weather fronts, storms, and other significant weather events.

Summary of Units:

```
T2M: °C (Degrees Celsius) - Average air temperature at 2 meters.
WS2M: m/s (Meters per second) - Wind speed at 2 meters.
PRECTOTCORR: mm/day (Millimeters per day) - Total precipitation.
RH2M: % (Percentage) - Relative humidity at 2 meters.
ALLSKY_SFC_SW_DWN: kWh/m²/day (Kilowatt-hours per square meter per day) -
Solar radiation.
T2M_MIN: °C - Minimum air temperature at 2 meters.
T2M_MAX: °C - Maximum air temperature at 2 meters.
PS: Pa (Pascals) - Surface pressure.
```

0.3 Write all the weather data to a .csv

[22]:
```
import requests
import pandas as pd
from io import StringIO
import os

# Define the district and its coordinates
district = "Bangalore"
coords = {"lat": 12.9716, "lon": 77.5946}

# Date range
start_date = "20200101"
end_date = "20201231"

# List of parameters to check
parameters = ["T2M", "WS2M", "PRECTOTCORR", "RH2M", "ALLSKY_SFC_SW_DWN",␣
 ↪"T2M_MIN", "T2M_MAX", "PS"]

# NASA POWER API URL template
url_template = "https://power.larc.nasa.gov/api/temporal/daily/point?
 ↪parameters={param}&community=AG&longitude={lon}&latitude={lat}&start={start}&end={end}&form

# Create directory if not exists
output_dir = "weather_data"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

# Dictionary to store data for each parameter
all_data = {}

# Loop through each parameter and fetch data
for param in parameters:
    # Construct the URL with the current parameter
    url = url_template.format(param=param, lon=coords["lon"],␣
 ↪lat=coords["lat"], start=start_date, end=end_date)
```

```
    # Send the API request
    response = requests.get(url)

    # Check the response status code
    if response.status_code == 200:
        try:
            # Load the response text into a pandas DataFrame (skip 9 header␣
 ↪lines)
            data = pd.read_csv(StringIO(response.text), skiprows=9)
            data.columns = [col.strip() for col in data.columns]  # Clean␣
 ↪column names

            # Keep the Year, DoY, and the parameter's values
            all_data[param] = data[['YEAR', 'DOY', data.columns[2]]]  # Store␣
 ↪only the relevant columns
            print(f"Parameter '{param}' is valid. Data fetched successfully.")
        except pd.errors.ParserError:
            print(f"Parameter '{param}' returned an error while parsing CSV.")
    else:
        print(f"Parameter '{param}' is invalid or returned an error. Status␣
 ↪code: {response.status_code}")

# Assuming all parameters have the same year and DoY, use the first DataFrame␣
 ↪for the Year and DoY columns
final_df = pd.DataFrame()
final_df['Year'] = all_data['T2M']['YEAR']  # Using the Year from one of the␣
 ↪dataframes
final_df['DayOfYear'] = all_data['T2M']['DOY']  # Using the Day of Year from␣
 ↪one of the dataframes

# Add parameter values as separate columns
for param, param_data in all_data.items():
    final_df[param] = param_data.iloc[:, 2].values  # Extract the parameter␣
 ↪values

# Save the final DataFrame to a CSV file
csv_filename = os.path.join(output_dir, f"{district}_weather_data.csv")
final_df.to_csv(csv_filename, index=False)
print(f"All parameters' data saved to {csv_filename}")
```

```
Parameter 'T2M' is valid. Data fetched successfully.
Parameter 'WS2M' is valid. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid. Data fetched successfully.
Parameter 'RH2M' is valid. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid. Data fetched successfully.
Parameter 'T2M_MIN' is valid. Data fetched successfully.
```

```
Parameter 'T2M_MAX' is valid. Data fetched successfully.
Parameter 'PS' is valid. Data fetched successfully.
All parameters' data saved to weather_data/Bangalore_weather_data.csv
```

0.4 Do this for all the districts now

[33]:
```python
import requests
import pandas as pd
from io import StringIO
import os

# Full list of Karnataka districts with coordinates
districts = {
    "Bagalkot": {"lat": 16.1867, "lon": 75.6964},
    "Bangalore": {"lat": 12.9716, "lon": 77.5946},
    "Bangalore Rural": {"lat": 13.18, "lon": 77.8},
    "Belgaum": {"lat": 15.8497, "lon": 74.4977},
    "Bellary": {"lat": 15.1394, "lon": 76.9214},
    "Bidar": {"lat": 17.9101, "lon": 77.5199},
    "Chamarajanagar": {"lat": 11.9234, "lon": 76.9395},
    "Chikkaballapur": {"lat": 13.435, "lon": 77.7315},
    "Chikmagalur": {"lat": 13.3184, "lon": 75.7723},
    "Chitradurga": {"lat": 14.2306, "lon": 76.398},
    "Dakshina Kannada": {"lat": 12.9141, "lon": 74.856},
    "Davanagere": {"lat": 14.4645, "lon": 75.9214},
    "Dharwad": {"lat": 15.3647, "lon": 75.1239},
    "Gadag": {"lat": 15.4327, "lon": 75.6245},
    "Gulbarga": {"lat": 17.3297, "lon": 76.8343},
    "Hassan": {"lat": 13.0072, "lon": 76.0945},
    "Haveri": {"lat": 14.7936, "lon": 75.4043},
    "Kodagu": {"lat": 12.3375, "lon": 75.8069},
    "Kolar": {"lat": 13.1337, "lon": 78.132},
    "Koppal": {"lat": 15.3478, "lon": 76.1548},
    "Mandya": {"lat": 12.524, "lon": 76.897},
    "Mysore": {"lat": 12.2958, "lon": 76.6394},
    "Raichur": {"lat": 16.2076, "lon": 77.3463},
    "Ramanagara": {"lat": 12.711, "lon": 77.2828},
    "Shimoga": {"lat": 13.9299, "lon": 75.5681},
    "Tumkur": {"lat": 13.34, "lon": 77.101},
    "Udupi": {"lat": 13.3409, "lon": 74.7421},
    "Uttara Kannada": {"lat": 14.8185, "lon": 74.1316},
    "Vijayanagara": {"lat": 15.22, "lon": 76.47},
    "Vijayapura": {"lat": 16.83, "lon": 75.71},
    "Yadgir": {"lat": 16.7709, "lon": 77.1375}
}
```

```
# Date range
start_date = "20200101"
end_date = "20231231"

# List of parameters to check
parameters = ["T2M", "WS2M", "PRECTOTCORR", "RH2M", "ALLSKY_SFC_SW_DWN",␣
 ↪"T2M_MIN", "T2M_MAX", "PS"]

# NASA POWER API URL template
url_template = "https://power.larc.nasa.gov/api/temporal/daily/point?
 ↪parameters={param}&community=AG&longitude={lon}&latitude={lat}&start={start}&end={end}&form

# Create directory if not exists
output_dir = "karnataka1_weather_data"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

# Fetch data for each district
for district, coords in districts.items():
    print(f"Processing data for {district}...")

    # Dictionary to store data for each parameter for this district
    all_data = {}

    # Loop through each parameter and fetch data
    for param in parameters:
        # Construct the URL with the current parameter
        url = url_template.format(param=param, lon=coords["lon"],␣
 ↪lat=coords["lat"], start=start_date, end=end_date)

        # Send the API request
        response = requests.get(url)

        # Check the response status code
        if response.status_code == 200:
            try:
                # Load the response text into a pandas DataFrame (skip 9 header␣
 ↪lines)
                data = pd.read_csv(StringIO(response.text), skiprows=9)
                data.columns = [col.strip() for col in data.columns]  # Clean␣
 ↪column names

                # Keep the Year, DoY, and the parameter's values
                all_data[param] = data[['YEAR', 'DOY', data.columns[2]]]  #␣
 ↪Store only the relevant columns
```

```
                print(f"Parameter '{param}' is valid for {district}. Data␣
↪fetched successfully.")
            except pd.errors.ParserError:
                print(f"Parameter '{param}' returned an error while parsing CSV␣
↪for {district}.")
        else:
            print(f"Parameter '{param}' is invalid or returned an error for␣
↪{district}. Status code: {response.status_code}")

    # Assuming all parameters have the same year and DoY, use the first␣
↪DataFrame for the Year and DoY columns
    final_df = pd.DataFrame()
    final_df['Year'] = all_data['T2M']['YEAR']  # Using the Year from one of␣
↪the dataframes
    final_df['DayOfYear'] = all_data['T2M']['DOY']  # Using the Day of Year␣
↪from one of the dataframes

    # Add parameter values as separate columns
    for param, param_data in all_data.items():
        final_df[param] = param_data.iloc[:, 2].values  # Extract the parameter␣
↪values

    # Save the final DataFrame to a CSV file for this district
    csv_filename = os.path.join(output_dir, f"{district}_weather_data.csv")
    final_df.to_csv(csv_filename, index=False)
    print(f"Data for {district} saved to {csv_filename}.\n")

print("All districts processed.")
```

```
Processing data for Bagalkot…
Parameter 'T2M' is valid for Bagalkot. Data fetched successfully.
Parameter 'WS2M' is valid for Bagalkot. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Bagalkot. Data fetched successfully.
Parameter 'RH2M' is valid for Bagalkot. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Bagalkot. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Bagalkot. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Bagalkot. Data fetched successfully.
Parameter 'PS' is valid for Bagalkot. Data fetched successfully.
Data for Bagalkot saved to karnataka1_weather_data/Bagalkot_weather_data.csv.

Processing data for Bangalore…
Parameter 'T2M' is valid for Bangalore. Data fetched successfully.
Parameter 'WS2M' is valid for Bangalore. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Bangalore. Data fetched successfully.
Parameter 'RH2M' is valid for Bangalore. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Bangalore. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Bangalore. Data fetched successfully.
```

```
Parameter 'T2M_MAX' is valid for Bangalore. Data fetched successfully.
Parameter 'PS' is valid for Bangalore. Data fetched successfully.
Data for Bangalore saved to karnataka1_weather_data/Bangalore_weather_data.csv.

Processing data for Bangalore Rural…
Parameter 'T2M' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'WS2M' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'RH2M' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Bangalore Rural. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Bangalore Rural. Data fetched successfully.
Parameter 'PS' is valid for Bangalore Rural. Data fetched successfully.
Data for Bangalore Rural saved to karnataka1_weather_data/Bangalore
Rural_weather_data.csv.

Processing data for Belgaum…
Parameter 'T2M' is valid for Belgaum. Data fetched successfully.
Parameter 'WS2M' is valid for Belgaum. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Belgaum. Data fetched successfully.
Parameter 'RH2M' is valid for Belgaum. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Belgaum. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Belgaum. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Belgaum. Data fetched successfully.
Parameter 'PS' is valid for Belgaum. Data fetched successfully.
Data for Belgaum saved to karnataka1_weather_data/Belgaum_weather_data.csv.

Processing data for Bellary…
Parameter 'T2M' is valid for Bellary. Data fetched successfully.
Parameter 'WS2M' is valid for Bellary. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Bellary. Data fetched successfully.
Parameter 'RH2M' is valid for Bellary. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Bellary. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Bellary. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Bellary. Data fetched successfully.
Parameter 'PS' is valid for Bellary. Data fetched successfully.
Data for Bellary saved to karnataka1_weather_data/Bellary_weather_data.csv.

Processing data for Bidar…
Parameter 'T2M' is valid for Bidar. Data fetched successfully.
Parameter 'WS2M' is valid for Bidar. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Bidar. Data fetched successfully.
Parameter 'RH2M' is valid for Bidar. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Bidar. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Bidar. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Bidar. Data fetched successfully.
Parameter 'PS' is valid for Bidar. Data fetched successfully.
```

```
Data for Bidar saved to karnataka1_weather_data/Bidar_weather_data.csv.

Processing data for Chamarajanagar…
Parameter 'T2M' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'WS2M' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'RH2M' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Chamarajanagar. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Chamarajanagar. Data fetched successfully.
Parameter 'PS' is valid for Chamarajanagar. Data fetched successfully.
Data for Chamarajanagar saved to
karnataka1_weather_data/Chamarajanagar_weather_data.csv.

Processing data for Chikkaballapur…
Parameter 'T2M' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'WS2M' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'RH2M' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Chikkaballapur. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Chikkaballapur. Data fetched successfully.
Parameter 'PS' is valid for Chikkaballapur. Data fetched successfully.
Data for Chikkaballapur saved to
karnataka1_weather_data/Chikkaballapur_weather_data.csv.

Processing data for Chikmagalur…
Parameter 'T2M' is valid for Chikmagalur. Data fetched successfully.
Parameter 'WS2M' is valid for Chikmagalur. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Chikmagalur. Data fetched successfully.
Parameter 'RH2M' is valid for Chikmagalur. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Chikmagalur. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Chikmagalur. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Chikmagalur. Data fetched successfully.
Parameter 'PS' is valid for Chikmagalur. Data fetched successfully.
Data for Chikmagalur saved to
karnataka1_weather_data/Chikmagalur_weather_data.csv.

Processing data for Chitradurga…
Parameter 'T2M' is valid for Chitradurga. Data fetched successfully.
Parameter 'WS2M' is valid for Chitradurga. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Chitradurga. Data fetched successfully.
Parameter 'RH2M' is valid for Chitradurga. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Chitradurga. Data fetched
successfully.
```

```
Parameter 'T2M_MIN' is valid for Chitradurga. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Chitradurga. Data fetched successfully.
Parameter 'PS' is valid for Chitradurga. Data fetched successfully.
Data for Chitradurga saved to
karnataka1_weather_data/Chitradurga_weather_data.csv.

Processing data for Dakshina Kannada…
Parameter 'T2M' is valid for Dakshina Kannada. Data fetched successfully.
Parameter 'WS2M' is valid for Dakshina Kannada. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Dakshina Kannada. Data fetched
successfully.
Parameter 'RH2M' is valid for Dakshina Kannada. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Dakshina Kannada. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Dakshina Kannada. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Dakshina Kannada. Data fetched successfully.
Parameter 'PS' is valid for Dakshina Kannada. Data fetched successfully.
Data for Dakshina Kannada saved to karnataka1_weather_data/Dakshina
Kannada_weather_data.csv.

Processing data for Davanagere…
Parameter 'T2M' is valid for Davanagere. Data fetched successfully.
Parameter 'WS2M' is valid for Davanagere. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Davanagere. Data fetched successfully.
Parameter 'RH2M' is valid for Davanagere. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Davanagere. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Davanagere. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Davanagere. Data fetched successfully.
Parameter 'PS' is valid for Davanagere. Data fetched successfully.
Data for Davanagere saved to
karnataka1_weather_data/Davanagere_weather_data.csv.

Processing data for Dharwad…
Parameter 'T2M' is valid for Dharwad. Data fetched successfully.
Parameter 'WS2M' is valid for Dharwad. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Dharwad. Data fetched successfully.
Parameter 'RH2M' is valid for Dharwad. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Dharwad. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Dharwad. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Dharwad. Data fetched successfully.
Parameter 'PS' is valid for Dharwad. Data fetched successfully.
Data for Dharwad saved to karnataka1_weather_data/Dharwad_weather_data.csv.

Processing data for Gadag…
Parameter 'T2M' is valid for Gadag. Data fetched successfully.
Parameter 'WS2M' is valid for Gadag. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Gadag. Data fetched successfully.
```

```
Parameter 'RH2M' is valid for Gadag. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Gadag. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Gadag. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Gadag. Data fetched successfully.
Parameter 'PS' is valid for Gadag. Data fetched successfully.
Data for Gadag saved to karnataka1_weather_data/Gadag_weather_data.csv.

Processing data for Gulbarga…
Parameter 'T2M' is valid for Gulbarga. Data fetched successfully.
Parameter 'WS2M' is valid for Gulbarga. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Gulbarga. Data fetched successfully.
Parameter 'RH2M' is valid for Gulbarga. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Gulbarga. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Gulbarga. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Gulbarga. Data fetched successfully.
Parameter 'PS' is valid for Gulbarga. Data fetched successfully.
Data for Gulbarga saved to karnataka1_weather_data/Gulbarga_weather_data.csv.

Processing data for Hassan…
Parameter 'T2M' is valid for Hassan. Data fetched successfully.
Parameter 'WS2M' is valid for Hassan. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Hassan. Data fetched successfully.
Parameter 'RH2M' is valid for Hassan. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Hassan. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Hassan. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Hassan. Data fetched successfully.
Parameter 'PS' is valid for Hassan. Data fetched successfully.
Data for Hassan saved to karnataka1_weather_data/Hassan_weather_data.csv.

Processing data for Haveri…
Parameter 'T2M' is valid for Haveri. Data fetched successfully.
Parameter 'WS2M' is valid for Haveri. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Haveri. Data fetched successfully.
Parameter 'RH2M' is valid for Haveri. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Haveri. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Haveri. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Haveri. Data fetched successfully.
Parameter 'PS' is valid for Haveri. Data fetched successfully.
Data for Haveri saved to karnataka1_weather_data/Haveri_weather_data.csv.

Processing data for Kodagu…
Parameter 'T2M' is valid for Kodagu. Data fetched successfully.
Parameter 'WS2M' is valid for Kodagu. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Kodagu. Data fetched successfully.
Parameter 'RH2M' is valid for Kodagu. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Kodagu. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Kodagu. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Kodagu. Data fetched successfully.
```

```
Parameter 'PS' is valid for Kodagu. Data fetched successfully.
Data for Kodagu saved to karnataka1_weather_data/Kodagu_weather_data.csv.

Processing data for Kolar…
Parameter 'T2M' is valid for Kolar. Data fetched successfully.
Parameter 'WS2M' is valid for Kolar. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Kolar. Data fetched successfully.
Parameter 'RH2M' is valid for Kolar. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Kolar. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Kolar. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Kolar. Data fetched successfully.
Parameter 'PS' is valid for Kolar. Data fetched successfully.
Data for Kolar saved to karnataka1_weather_data/Kolar_weather_data.csv.

Processing data for Koppal…
Parameter 'T2M' is valid for Koppal. Data fetched successfully.
Parameter 'WS2M' is valid for Koppal. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Koppal. Data fetched successfully.
Parameter 'RH2M' is valid for Koppal. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Koppal. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Koppal. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Koppal. Data fetched successfully.
Parameter 'PS' is valid for Koppal. Data fetched successfully.
Data for Koppal saved to karnataka1_weather_data/Koppal_weather_data.csv.

Processing data for Mandya…
Parameter 'T2M' is valid for Mandya. Data fetched successfully.
Parameter 'WS2M' is valid for Mandya. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Mandya. Data fetched successfully.
Parameter 'RH2M' is valid for Mandya. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Mandya. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Mandya. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Mandya. Data fetched successfully.
Parameter 'PS' is valid for Mandya. Data fetched successfully.
Data for Mandya saved to karnataka1_weather_data/Mandya_weather_data.csv.

Processing data for Mysore…
Parameter 'T2M' is valid for Mysore. Data fetched successfully.
Parameter 'WS2M' is valid for Mysore. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Mysore. Data fetched successfully.
Parameter 'RH2M' is valid for Mysore. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Mysore. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Mysore. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Mysore. Data fetched successfully.
Parameter 'PS' is valid for Mysore. Data fetched successfully.
Data for Mysore saved to karnataka1_weather_data/Mysore_weather_data.csv.

Processing data for Raichur…
```

```
Parameter 'T2M' is valid for Raichur. Data fetched successfully.
Parameter 'WS2M' is valid for Raichur. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Raichur. Data fetched successfully.
Parameter 'RH2M' is valid for Raichur. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Raichur. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Raichur. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Raichur. Data fetched successfully.
Parameter 'PS' is valid for Raichur. Data fetched successfully.
Data for Raichur saved to karnataka1_weather_data/Raichur_weather_data.csv.

Processing data for Ramanagara…
Parameter 'T2M' is valid for Ramanagara. Data fetched successfully.
Parameter 'WS2M' is valid for Ramanagara. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Ramanagara. Data fetched successfully.
Parameter 'RH2M' is valid for Ramanagara. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Ramanagara. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Ramanagara. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Ramanagara. Data fetched successfully.
Parameter 'PS' is valid for Ramanagara. Data fetched successfully.
Data for Ramanagara saved to
karnataka1_weather_data/Ramanagara_weather_data.csv.

Processing data for Shimoga…
Parameter 'T2M' is valid for Shimoga. Data fetched successfully.
Parameter 'WS2M' is valid for Shimoga. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Shimoga. Data fetched successfully.
Parameter 'RH2M' is valid for Shimoga. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Shimoga. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Shimoga. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Shimoga. Data fetched successfully.
Parameter 'PS' is valid for Shimoga. Data fetched successfully.
Data for Shimoga saved to karnataka1_weather_data/Shimoga_weather_data.csv.

Processing data for Tumkur…
Parameter 'T2M' is valid for Tumkur. Data fetched successfully.
Parameter 'WS2M' is valid for Tumkur. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Tumkur. Data fetched successfully.
Parameter 'RH2M' is valid for Tumkur. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Tumkur. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Tumkur. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Tumkur. Data fetched successfully.
Parameter 'PS' is valid for Tumkur. Data fetched successfully.
Data for Tumkur saved to karnataka1_weather_data/Tumkur_weather_data.csv.

Processing data for Udupi…
Parameter 'T2M' is valid for Udupi. Data fetched successfully.
Parameter 'WS2M' is valid for Udupi. Data fetched successfully.
```

```
Parameter 'PRECTOTCORR' is valid for Udupi. Data fetched successfully.
Parameter 'RH2M' is valid for Udupi. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Udupi. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Udupi. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Udupi. Data fetched successfully.
Parameter 'PS' is valid for Udupi. Data fetched successfully.
Data for Udupi saved to karnataka1_weather_data/Udupi_weather_data.csv.

Processing data for Uttara Kannada…
Parameter 'T2M' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'WS2M' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'RH2M' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Uttara Kannada. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Uttara Kannada. Data fetched successfully.
Parameter 'PS' is valid for Uttara Kannada. Data fetched successfully.
Data for Uttara Kannada saved to karnataka1_weather_data/Uttara
Kannada_weather_data.csv.

Processing data for Vijayanagara…
Parameter 'T2M' is valid for Vijayanagara. Data fetched successfully.
Parameter 'WS2M' is valid for Vijayanagara. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Vijayanagara. Data fetched successfully.
Parameter 'RH2M' is valid for Vijayanagara. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Vijayanagara. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Vijayanagara. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Vijayanagara. Data fetched successfully.
Parameter 'PS' is valid for Vijayanagara. Data fetched successfully.
Data for Vijayanagara saved to
karnataka1_weather_data/Vijayanagara_weather_data.csv.

Processing data for Vijayapura…
Parameter 'T2M' is valid for Vijayapura. Data fetched successfully.
Parameter 'WS2M' is valid for Vijayapura. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Vijayapura. Data fetched successfully.
Parameter 'RH2M' is valid for Vijayapura. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Vijayapura. Data fetched
successfully.
Parameter 'T2M_MIN' is valid for Vijayapura. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Vijayapura. Data fetched successfully.
Parameter 'PS' is valid for Vijayapura. Data fetched successfully.
Data for Vijayapura saved to
karnataka1_weather_data/Vijayapura_weather_data.csv.

Processing data for Yadgir…
```

```
Parameter 'T2M' is valid for Yadgir. Data fetched successfully.
Parameter 'WS2M' is valid for Yadgir. Data fetched successfully.
Parameter 'PRECTOTCORR' is valid for Yadgir. Data fetched successfully.
Parameter 'RH2M' is valid for Yadgir. Data fetched successfully.
Parameter 'ALLSKY_SFC_SW_DWN' is valid for Yadgir. Data fetched successfully.
Parameter 'T2M_MIN' is valid for Yadgir. Data fetched successfully.
Parameter 'T2M_MAX' is valid for Yadgir. Data fetched successfully.
Parameter 'PS' is valid for Yadgir. Data fetched successfully.
Data for Yadgir saved to karnataka1_weather_data/Yadgir_weather_data.csv.

All districts processed.
```

The spatial resolution of NASA POWER meteorological data, is derived from the MERRA-2 assimilation model, which has a resolution of 0.5° x 0.625° latitude/longitude. This translates to roughly 55 km by 69 km at the equator. The data is available globally on this grid, making it useful for broad, regional analyses but not detailed, local-scale studies.

For solar radiation parameters, data from the CERES and GEWEX projects are similarly provided at a 1° x 1° resolution, meaning each grid cell covers around 111 km x 111 km at the equator.

Thus, while the data is highly useful for regional or district-level analysis, it may not fully capture microclimates or very localized weather patterns due to its grid-based resolution.

For more details, you can refer to NASA POWER's official documentation and FAQs.

To determine the mapping between a district's latitude and longitude and the corresponding grid cell center from NASA POWER data, you need to understand how the grid system is structured and identify the center of the grid cells based on their resolution. Steps to Map District Latitude/Longitude to Grid Cells:

```
Determine the Grid Resolution:
    Latitude resolution is 0.5°.
    Longitude resolution is 0.625°.

This means each grid cell spans 0.5° in latitude and 0.625° in longitude.

Identify the Grid Cell Center:
    The grid cell center can be calculated by rounding the given latitude and longitude to the

Example Calculation:
    Suppose a district has a latitude of 12.97° and longitude of 77.59°.
        Round 12.97° to the nearest 0.5° multiple: This would be 13.0°.
        Round 77.59° to the nearest 0.625° multiple: This would be 77.5°.
    The grid cell center corresponding to this district would be 13.0° latitude and 77.5° longi

Check Grid Boundaries:
    Each grid cell spans from its center to half the grid resolution around it. For a center at

Automated Tools for Mapping:
    You can automate this process by writing a simple script that rounds the district latitude
```

To determine how many distinct grid cells Karnataka can be divided into using NASA POWER data, we need to consider the spatial resolution of the data.

```
Spatial Resolution: NASA POWER data has a resolution of 0.5° latitude by 0.625° longitude, whi

Area of Karnataka: Karnataka has a total area of approximately 191,791 square kilometers.

Size of Each Grid Cell: At the latitude of Karnataka (roughly between 11.6° N and 18.5° N), th
```

Calculation:

```
Area of Karnataka: 191,791 km²
Area of each grid cell: ~3,500 km² (an approximation between the latitude and longitude values)
```

Thus, Karnataka can be divided into approximately: 191,791 km23,500 km2 55 grid cells 3,500 km2191,791 km2 55 grid cells Conclusion:

Karnataka would be divided into approximately 55 distinct grid cells at NASA POWER's spatial resolution. Each grid cell would represent a region covering an area between 3,400 km² and 3,800 km². The exact number of grid cells could vary depending on the exact boundaries and latitudinal/longitudinal extent.

There are 31 districts in karnataka less than the 55 grid cells. Which provide a fair degree of spatial resolution

[32]:
```python
# Define the coordinates for all the districts in Karnataka
districts = {
    "Bagalkot": {"lat": 16.1867, "lon": 75.6964},
    "Bangalore": {"lat": 12.9716, "lon": 77.5946},
    "Bangalore Rural": {"lat": 13.18, "lon": 77.8},
    "Belgaum": {"lat": 15.8497, "lon": 74.4977},
    "Bellary": {"lat": 15.1394, "lon": 76.9214},
    "Bidar": {"lat": 17.9101, "lon": 77.5199},
    "Chamarajanagar": {"lat": 11.9234, "lon": 76.9395},
    "Chikkaballapur": {"lat": 13.435, "lon": 77.7315},
    "Chikmagalur": {"lat": 13.3184, "lon": 75.7723},
    "Chitradurga": {"lat": 14.2306, "lon": 76.398},
    "Dakshina Kannada": {"lat": 12.9141, "lon": 74.856},
    "Davanagere": {"lat": 14.4645, "lon": 75.9214},
    "Dharwad": {"lat": 15.3647, "lon": 75.1239},
    "Gadag": {"lat": 15.4327, "lon": 75.6245},
    "Gulbarga": {"lat": 17.3297, "lon": 76.8343},
    "Hassan": {"lat": 13.0072, "lon": 76.0945},
    "Haveri": {"lat": 14.7936, "lon": 75.4043},
    "Kodagu": {"lat": 12.3375, "lon": 75.8069},
    "Kolar": {"lat": 13.1337, "lon": 78.132},
    "Koppal": {"lat": 15.3478, "lon": 76.1548},
    "Mandya": {"lat": 12.524, "lon": 76.897},
    "Mysore": {"lat": 12.2958, "lon": 76.6394},
    "Raichur": {"lat": 16.2076, "lon": 77.3463},
    "Ramanagara": {"lat": 12.711, "lon": 77.2828},
```

```
    "Shimoga": {"lat": 13.9299, "lon": 75.5681},
    "Tumkur": {"lat": 13.34, "lon": 77.101},
    "Udupi": {"lat": 13.3409, "lon": 74.7421},
    "Uttara Kannada": {"lat": 14.8185, "lon": 74.1316},
    "Vijayanagara": {"lat": 15.22, "lon": 76.47},
    "Vijayapura": {"lat": 16.83, "lon": 75.71},
    "Yadgir": {"lat": 16.7709, "lon": 77.1375}
}

# Function to map district lat/lon to nearest grid center
def map_to_grid(lat, lon):
    # Round latitude to nearest 0.5 degree multiple
    grid_lat = round(lat * 2) / 2

    # Round longitude to nearest 0.625 degree multiple
    grid_lon = round(lon / 0.625) * 0.625

    return grid_lat, grid_lon

# Create a dictionary to store grid centers for all districts
district_grid_centers = {district: map_to_grid(coords['lat'], coords['lon'])␣
 ↪for district, coords in districts.items()}

# Display the grid centers
district_grid_centers
```

[32]:
```
{'Bagalkot': (16.0, 75.625),
 'Bangalore': (13.0, 77.5),
 'Bangalore Rural': (13.0, 77.5),
 'Belgaum': (16.0, 74.375),
 'Bellary': (15.0, 76.875),
 'Bidar': (18.0, 77.5),
 'Chamarajanagar': (12.0, 76.875),
 'Chikkaballapur': (13.5, 77.5),
 'Chikmagalur': (13.5, 75.625),
 'Chitradurga': (14.0, 76.25),
 'Dakshina Kannada': (13.0, 75.0),
 'Davanagere': (14.5, 75.625),
 'Dharwad': (15.5, 75.0),
 'Gadag': (15.5, 75.625),
 'Gulbarga': (17.5, 76.875),
 'Hassan': (13.0, 76.25),
 'Haveri': (15.0, 75.625),
 'Kodagu': (12.5, 75.625),
 'Kolar': (13.0, 78.125),
 'Koppal': (15.5, 76.25),
 'Mandya': (12.5, 76.875),
```

```
'Mysore': (12.5, 76.875),
'Raichur': (16.0, 77.5),
'Ramanagara': (12.5, 77.5),
'Shimoga': (14.0, 75.625),
'Tumkur': (13.5, 76.875),
'Udupi': (13.5, 75.0),
'Uttara Kannada': (15.0, 74.375),
'Vijayanagara': (15.0, 76.25),
'Vijayapura': (17.0, 75.625),
'Yadgir': (17.0, 76.875)}
```