Spark Lab Task: Climate and Environmental Data Analysis

Task Overview

In this lab, you will implement a Spark application using RDD-based transformations and aggregations to analyze real-world climate observations from the **NOAA Global Surface Summary of the Day (GSOD)** dataset. The goal is to practice data parsing, cleaning, transformation, and large-scale aggregation using Spark's RDD API.

Dataset

Use the NOAA GSOD dataset, which contains daily climate measurements for weather stations around the world. Each record includes information such as temperature, pressure, wind speed, precipitation, and extreme event indicators.

A typical CSV header contains the following fields:

```
"STATION","DATE","LATITUDE","LONGITUDE","ELEVATION","NAME",
"TEMP","TEMP_ATTRIBUTES","DEWP","DEWP_ATTRIBUTES",
"SLP","SLP_ATTRIBUTES","STP","STP_ATTRIBUTES",
"VISIB","VISIB_ATTRIBUTES","WDSP","WDSP_ATTRIBUTES",
"MXSPD","GUST","MAX","MAX_ATTRIBUTES","MIN","MIN_ATTRIBUTES",
"PRCP","PRCP_ATTRIBUTES","SNDP","FRSHTT"
```

Dataset link:

- NOAA GSOD: `https://www.ncei.noaa.gov/data/global-summary-of-the-day/` (i.e. Year 2025 direct link)

Tasks

1. Data Loading

- Load one or more (one year) GSOD CSV files into an RDD.
- Examine the raw data format and determine how to parse each column of interest.

2. Data Cleaning

Extract and clean the following fields: `station_id` (`STATION`), `date` (`DATE`), `temp_avg` (`TEMP`), `temp_max` (`MAX`), `temp_min` (`MIN`), `precipitation` (`PRCP`), `wind_speed_avg` (`WDSP`), `wind_gust_max` (`GUST`), `extreme_events` (`FRSHTT`)

Cleaning steps:

- Remove records with missing or invalid numeric fields (e.g., `9999.9`).
- Convert temperature, precipitation, and wind speed values to numeric types.
- Convert `DATE` into **year**, **month**, and **season**.

3. Data Transformation

- Map cleaned data to key–value pairs suitable for aggregation.
- For seasonal grouping, assign each date to: Winter, Spring, Summer, or Autumn.
- Extract extreme event flags (Fog, Rain, Snow, Hail, Thunder, Tornado) from `FRSHTT`.

4. Aggregations and Climate Analysis

Using RDD transformations and actions, compute:

- Monthly and yearly average temperatures for each station.
- Long-term temperature trends (warming or cooling) per station.
- Seasonal precipitation averages.
- Stations with the highest maximum daily temperatures.
- Detection of extreme events:
  - Days with extreme heat (very high `TEMP` or `MAX`).
  - High-wind days (`MXSPD` or `GUST`).
  - Rain, snow, or thunderstorms indicated by `FRSHTT`.

5. Saving Results

- Store all aggregated outputs in the file system as `csv` or `txt`.
- Include summary statistics, such as:
  - Hottest year on record (based on annual averages).
  - Wettest station or region (based on precipitation).
  - Station with the highest recorded wind gust.

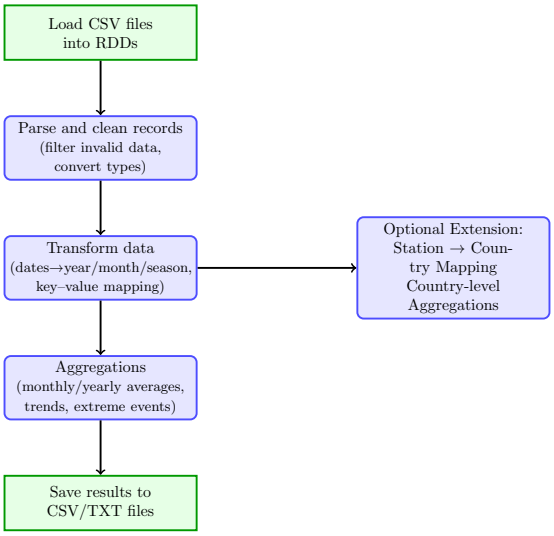

Figure 1: Workflow for the Spark RDD Climate Data Processing Task

Optional Extension (Bonus)

Extension 1: Country lookup

- Use a station lookup file (e.g., NOAA station metadata) to map:

  `STATION` → `Country / Region / WMO Code`

- Aggregate climate statistics at the country level.
- Identify countries most affected by:
  - Increasing temperatures,
  - Heavy rainfall,
  - Wind extremes,
  - Frequent extreme weather events (`FRSHTT`).

Extension 2: Spark DataFrames and Spark SQL

While the main objective of the assignment is to practice Spark **RDD**-based data processing, students may optionally extend their solution by implementing part of the analysis using **Spark DataFrames** or **Spark SQL**. This extension can include:

- Loading the GSOD dataset using `spark.read.csv()` with an explicit schema.
- Performing data cleaning with DataFrame operations such as `withColumn()`, `filter()`, and `dropna()`.
- Computing monthly or yearly aggregates using `groupBy()` and SQL-style functions.
- Registering the cleaned DataFrame as a temporary view and expressing part of the analysis using SQL queries:
  - average temperatures per month,
  - seasonal precipitation,
  - identification of extreme events,
  - long-term warming trends.
- Comparing execution plans between the RDD and DataFrame/SQL versions using `explain()` to observe the optimizations performed by Catalyst.

This extension is voluntary but recommended for students who wish to gain experience with the modern higher-level Spark APIs and understand how the DataFrame and SQL execution engine differs from RDD-based processing.

Expected Output

Your final submission should include:

- Spark code with documentation.
- Output files with all aggregated results.
- A short report discussing:
  - Temperature trends,
  - Seasonal precipitation patterns,
  - Extreme event occurrences.