ClimateScope — Final Project Report

Executive Summary

ClimateScope is a Streamlit-based climate intelligence dashboard that ingests cleaned weather data, provides interactive filtering, computes derived metrics, and offers visual analytics for decision making. This report documents methodology, visualization types, sample insights, and recommendations for future enhancements.

Methodology

1. Data ingestion: CSV upload or environment-specified path. Supports large CSVs with safe parsing of timestamp columns. 2. Preprocessing: parse `last_updated`, normalize to `date`, extract `month`, `season`, and ensure numeric coercion for key metrics (temperature, humidity, precipitation, wind). 3. Feature engineering: computed `heat_index`, `wind_chill`, and per-country `temp_7day_avg` (rolling 7-day mean). 4. Filtering & aggregation: interactive filters (countries, date range, metric, normalization) applied before visualization. Aggregation modes: daily, monthly, seasonal where applicable. 5. Statistical checks: basic summary statistics, distribution checks, correlation matrix for core metrics. 6. Geospatial handling: sampling for heavy map loads, scatter-geo and choropleth for different granularities.

Visualizations

Included visualization types and purpose: - KPI Tiles: quick metrics (avg temperature, rainfall variability, extreme event risk) for executive overview. - Scatter Geo: city-level mapping of chosen metric to spot geographic clusters. - Choropleth Map: country-level averages to reveal macro patterns. - Time Series (Line): temporal trends per-country for selected metric. -Comparative Bars: recent vs historical comparisons (last 30 days vs baseline). - Seasonal Bars: grouped seasonal analysis (Winter/Spring/Summer/Autumn). - Scatter & Box: correlation (temperature vs humidity) and spread/outlier detection. - Histogram & Distribution: frequency and concentration detection with marginal boxplots. - Correlation Heatmap: pairwise relationships between core metrics. - Extreme Events Dashboards: timeline scatter, hotspot maps, seasonal counts, and full event tables.

Key Insights — How to read results

- High avg temperature combined with high humidity and many threshold exceedances indicates heat stress risk; prioritize regions with repeated extreme hits. - Choropleth + time series: countries with rising trends vs stable baselines are candidates for intervention or deeper study. -Seasonal bars reveal vulnerability windows; align resource allocation (cooling, water resources) to seasonal peaks. - Correlation heatmap helps identify dependent variables to include in predictive models (e.g., temperature-humidity relationships). - Extreme Events timeline and geographic clustering detect spatial-temporal hotspots for targeted monitoring or alerts. Note: Exact numeric insights require applying filters in the dashboard; this report describes how to interpret outputs rather than giving single-number conclusions.

Considerations for Future Enhancements

1. Data pipeline & automation - Implement ETL: scheduled ingestion from APIs (NOAA, ECMWF) and a database (Postgres). Add data validation & monitoring. 2. Performance & scaling - Use server-side caching for heavy aggregations, deploy behind Gunicorn + Nginx. Consider vectorized geospatial indexes or PostGIS for maps. 3. UI/UX upgrades - Dark mode toggle, animated KPI counters, mobile-responsive layout, and accessibility improvements (ARIA labels, keyboard navigation). 4. Advanced analytics - Integrate anomaly detection (time-series models), forecasting (ARIMA/Prophet), and clustering (DBSCAN) for hotspot detection. 5. Alerts & automation - Add scheduled reports, email/push alerts on threshold breaches, and webhook integrations for stakeholders. 6. Geospatial improvements - Use tile servers, heatmaps, and clustering on maps for dense datasets. Add administrative boundary overlays. 7. Reproducibility & testing - Add unit tests for preprocessing steps, CI/CD pipeline, versioned datasets, and data contracts. 8. Security & deployment - Secure file uploads, rate limit endpoints, and sanitize inputs. Deploy on a managed cloud with TLS and auth (OAuth / SSO).

How to run the Dashboard ■

1. Install: `pip install -r requirements.txt` (Streamlit, pandas, plotly). 2. Place data at `data/GlobalWeatherRepository_cleaned.csv` OR use the sidebar CSV uploader. 3. Run: `streamlit run app.py` and open the local URL shown by Streamlit. 4. For production: containerize (Docker), attach a persistent DB, and deploy to a cloud provider (Render, Railway, or AWS).