From Acquisition to Validation: An Integrative Workshop on Climate Data Processing with Earth Engine

Saurav Bhattarai*

May 16, 2025

Abstract

This document presents a comprehensive workshop framework for climate data processing, integrating three open-source Python tools developed by Jackson State University Water Resources Lab. The workshop takes participants through a complete workflow spanning data acquisition, analysis, and validation. GeoClimate-Fetcher provides an intuitive interface for downloading Google Earth Engine climate data. Index-Visualizer transforms this raw data into standardized climate indices with advanced visualization capabilities. GeeData-GroundData-validator enables rigorous validation of gridded products against ground observations. Together, these tools form a coherent pipeline that addresses the challenges of working with climate data at each stage of the research process. The workshop is designed for researchers, water resource managers, and climate scientists seeking to enhance their climate data analysis capabilities without extensive programming expertise. This document provides organizers with a structured workshop plan, covering installation requirements, teaching materials, hands-on exercises, and a culminating case study that demonstrates the integrated workflow.

Contents

*Department of Civil and Environmental Engineering, Jackson State University, Jackson, USA. Email: saurav.bhattarai@students.jsums.edu

1 Introduction

1.1 The Challenge of Climate Data Processing

Climate research and water resources management increasingly rely on large-scale spatiotemporal datasets. Working with climate data presents several key challenges:

- **Data Acquisition:** Retrieving climate data often requires specialized programming skills, particularly when accessing large-scale datasets from platforms like Google Earth Engine.
- **Analysis:** Converting raw climate variables into meaningful indices requires knowledge of climate science and significant computational effort.
- **Validation:** Gridded climate products must be validated against ground observations to assess their reliability for specific regions and applications.

While many researchers have expertise in one of these domains, few possess the comprehensive skill set needed to implement the full workflow from acquisition to validation. This knowledge gap frequently leads to inefficient workflows, incomplete analyses, or over-reliance on datasets that may not be optimal for particular research questions.

1.2 Workshop Overview

This workshop introduces an integrated solution through three complementary open-source Python tools that collectively address the complete climate data workflow:

1. **GeoClimate-Fetcher:** An intuitive package for downloading climate data from Google Earth Engine
2. **Index-Visualizer:** A tool for calculating and visualizing standardized climate indices
3. **GeeData-GroundData-validator:** A comprehensive application for validating gridded climate products against ground observations

Together, these tools form a logical progression that guides users from data acquisition through analysis to validation, creating a seamless workflow for climate data processing. Figure 1 illustrates this integrated approach. The workshop is designed to be conducted over a one or two-day period, with each tool covered in a dedicated section. Participants will progress through the complete workflow, starting with data acquisition and ending with comprehensive validation and product selection.

1.3 Target Audience

This workshop is designed for:

- **Hydrologists and water resource engineers** seeking to incorporate climate data into hydrologic models

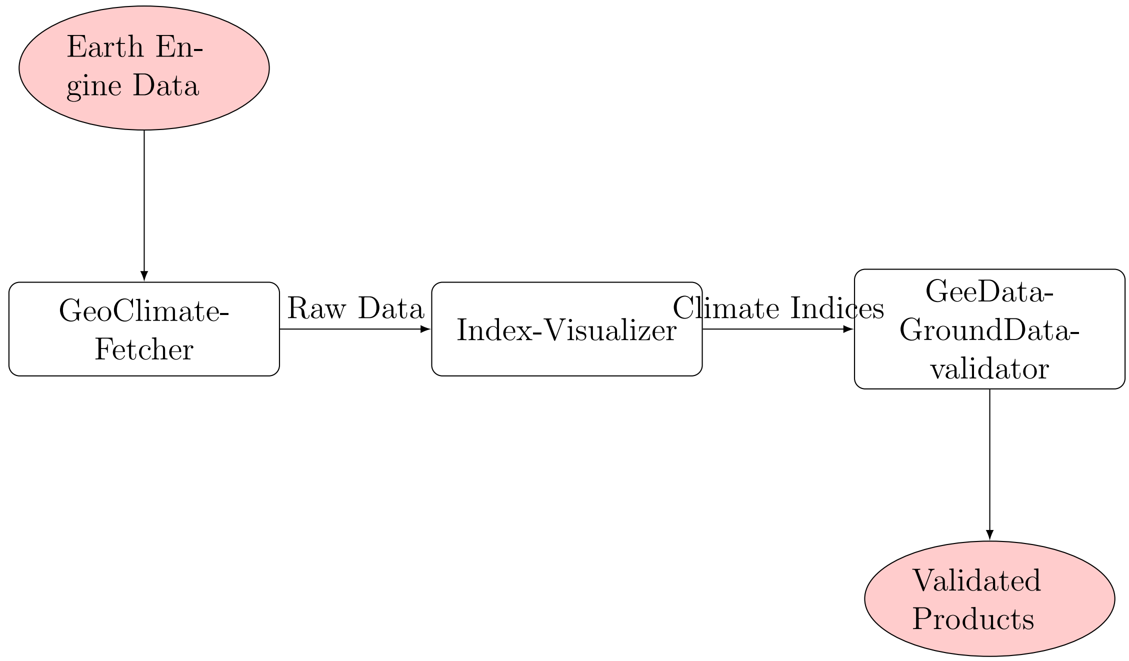

Figure 1: Integrated climate data workflow using the three tools

- **Climate scientists** who need efficient tools for analyzing and validating climate datasets
- **Environmental researchers** working on climate impacts in diverse geographical contexts
- **Graduate students** developing research projects related to climate or hydrology
- **GIS specialists** looking to enhance their climate data processing capabilities

Participants should have basic familiarity with Python programming, but extensive expertise is not required as the tools provide graphical interfaces that abstract much of the underlying complexity.

2 Workshop Prerequisites and Setup

2.1 Technical Requirements

To participate in the workshop, attendees will need:

- **Computer Requirements:**
  - Windows, macOS, or Linux operating system
  - Minimum 8 GB RAM (16 GB recommended)
  - 10 GB free disk space

- Administrator privileges to install software

- **Software Prerequisites:**
  - Python 3.8 to 3.12 (not 3.13)
  - Git version control system
  - Visual Studio Code (recommended) or another Python IDE
- **Accounts and Access:**
  - Google Earth Engine account (must be approved before workshop)
  - Google Cloud Project with Earth Engine API enabled

2.2 Pre-Workshop Setup

Participants should complete the following steps before attending the workshop:

1. **Python Installation:**
   - Download and install Python 3.8-3.12 from https://www.python.org/downloads/
   - Verify installation by running `python --version` in a terminal
2. **Earth Engine Account:**
   - Sign up at https://earthengine.google.com/signup/
   - Note: Approval may take several days, so register well in advance
3. **Git Installation:**
   - Download and install from https://git-scm.com/downloads
   - Verify with `git --version`
4. **IDE Setup:**
   - Install Visual Studio Code from https://code.visualstudio.com/download
   - Install the Python extension from the VS Code marketplace

2.3 Installation of Workshop Tools

2.3.1 Recommended: Using venv (pip)

GeoClimate-Fetcher Installation

```
# Clone the repository
git clone https://github.com/Saurav-JSU/GeoClimate-Fetcher.git
cd GeoClimate-Fetcher

# Create and activate a virtual environment
python -m venv .venv

# On Windows:
.venv\Scripts\activate
# On macOS/Linux:
source .venv/bin/activate

# Upgrade pip
python -m pip install --upgrade pip

# Install the package in development mode
pip install -e .

# Authenticate with Google Earth Engine
earthengine authenticate
```

Index-Visualizer Installation

```
# Clone the repository
git clone https://github.com/Saurav-JSU/Index-Visualizer.git
cd Index-Visualizer

# Create and activate a virtual environment
python -m venv climate-venv

# On Windows:
climate-venv\Scripts\activate
# On macOS/Linux:
source climate-venv/bin/activate

# Install required packages
pip install -r requirements.txt

# Launch the application
jupyter notebook main.ipynb
```

GeeData-GroundData-validator Installation

```
# Clone the repository
git clone https://github.com/Saurav-JSU/GeeData-GroundData-validator.git
cd GeeData-GroundData-validator

# Create and activate a virtual environment
python -m venv venv

# On Windows:
venv\Scripts\activate
# On macOS/Linux:
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt

# Launch the application
python main.py
```

2.3.2 Alternative: Using Conda

Using Conda (Alternative)

For users who prefer Conda, the tools can be installed using the following alternative approach:

```
# For GeoClimate-Fetcher
conda env create -f geoclimate_fetcher/environment.yml
conda activate geoclimate-fetcher

# For Index-Visualizer
conda env create -f environment.yml
conda activate climate-analysis

# For GeeData-GroundData-validator
conda create -n climate-data-fetcher python=3.8
conda activate climate-data-fetcher
pip install -r requirements.txt
```

Conda is particularly helpful for managing complex dependencies, especially geospatial packages like GDAL and GeoPandas which can be challenging to install with pip alone.

2.4 Troubleshooting Common Installation Issues

- **Permission Error During Installation:**
  - If you encounter an access denied error during installation, simply run the install command again
  - For persistent issues, add the `--user` flag: `pip install --user -e .`
- **Earth Engine Authentication Failure:**
  - Ensure your GEE account is approved
  - Run `earthengine authenticate` separately and follow the prompts
  - Verify your Google Cloud Project has the Earth Engine API enabled
- **Dependency Installation Problems:**
  - For issues with geospatial packages, consider using Conda instead of pip
  - On Linux, you may need additional system packages:
    `sudo apt-get install python3-dev libgdal-dev libproj-dev libgeos-dev`
  - On macOS, try: `brew install gdal geos proj`
- **PyQt5 Installation Issues:**

- On Linux: `sudo apt-get install python3-pyqt5`
- On macOS: `brew install qt`

2.5 Verifying Installations

After installing each tool, participants should verify their installation with these simple checks:

Installation Verification Steps

- **GeoClimate-Fetcher:**

```
# From the GeoClimate-Fetcher directory
jupyter notebook geoclimate_fetcher/notebooks/interactive_gui.ipynb
# Verify that the interactive GUI launches successfully
```

- **Index-Visualizer:**

```
# From the Index-Visualizer directory
python -c "import ee; ee.Initialize(); print('Earth Engine initialized')"
# Should print "Earth Engine initialized" without errors
```

- **GeeData-GroundData-validator:**

```
# From the GeeData-GroundData-validator directory
python -c "from PyQt5 import QtWidgets; print('PyQt5 installed')"
# Should print "PyQt5 installed" without errors
```

2.6 Workshop Environment Preparation

Organizers should prepare the following to ensure a smooth workshop experience:

- Stable internet connection with sufficient bandwidth for all participants
- Pre-download dataset examples to reduce reliance on internet connectivity
- USB drives with installation files for all required software
- Printed troubleshooting guides for quick reference
- Pre-workshop online session to verify participant setup
- Teaching assistants designated for installation support

3 Part 1: Data Acquisition with GeoClimate-Fetcher

3.1 Introduction to GeoClimate-Fetcher

GeoClimate-Fetcher serves as the entry point to the climate data workflow, providing an intuitive interface for downloading a variety of climate datasets from Google Earth Engine. This tool addresses the common challenges researchers face when acquiring climate data:

- The need for specialized programming skills to access Earth Engine
- Difficulty in defining precise study areas
- Challenges in navigating the vast catalog of available datasets
- Complexities in configuring data extraction and download options

By providing a graphical user interface for these tasks, GeoClimate-Fetcher democratizes access to climate data, making it available to researchers regardless of their programming expertise.

3.2 Core Functionality and Workflow

Participants will explore the core functionality of GeoClimate-Fetcher through a guided walkthrough of its five key components:

3.2.1 Google Earth Engine Authentication

The authentication interface provides:

- Project ID configuration
- Secure credential storage
- Authentication status monitoring
- Connection troubleshooting

Participants will learn how to properly configure their Earth Engine credentials and understand how these credentials are managed securely across sessions.

3.2.2 Area of Interest Selection

GeoClimate-Fetcher offers three flexible methods for defining study areas:

- Drawing polygons directly on an interactive map
- Uploading shapefiles or GeoJSON files
- Entering precise geographic coordinates

This section includes hands-on exercises where participants define several study areas using different methods, understanding the trade-offs between precision and convenience for each approach.

3.2.3 Dataset Selection and Exploration

Participants will explore GeoClimate-Fetcher's extensive catalog of 54+ climate datasets across six categories:

- Precipitation (11 datasets)
- Temperature (10 datasets)
- Soil Moisture (8 datasets)
- Evapotranspiration (17 datasets)
- NDVI (5 datasets)
- Digital Elevation Models (3 datasets)

Exercises will include searching for specific datasets, examining dataset metadata, and understanding the characteristics of different data sources.

3.2.4 Band Selection and Time Range Configuration

This section covers:

- Identifying available bands within selected datasets
- Selecting specific variables for download
- Configuring precise time periods
- Understanding temporal resolution considerations

Participants will practice selecting appropriate bands and time ranges for different research scenarios, learning how these choices affect data volume and processing time.

3.2.5 Download Configuration and Execution

The final component focuses on configuring and executing data downloads:

- Selecting extraction modes (time-series averages vs. gridded data)
- Choosing between local downloads and Google Drive exports
- Configuring output formats (CSV, GeoTIFF, NetCDF)
- Managing download progress and troubleshooting

Hands-on exercises will guide participants through downloading climate data for the workshop's case study region, which will be used throughout the remaining sections.

3.3 Hands-On Exercise: Regional Climate Data Acquisition

Participants will apply their knowledge in a comprehensive exercise:

1. Select a watershed of interest using the drawing tool
2. Choose precipitation and temperature datasets from two different sources
3. Configure a 10-year time period (2010-2020)
4. Select appropriate bands for each dataset
5. Download both time-series and gridded data for subsequent analysis

This exercise establishes the foundation for the remainder of the workshop, as the downloaded data will serve as input for the Index-Visualizer tool in the next section.

3.4 Advanced Features and Customization

For participants with more advanced needs, this section covers:

- Handling larger-than-memory datasets
- Working with very high-resolution data
- Customizing projection systems
- Batch processing multiple regions
- Integrating GeoClimate-Fetcher into custom Python workflows

3.5 Transition to Analysis

As a bridge to the next section, participants will organize their downloaded data and prepare it for analysis with Index-Visualizer. This includes:

- Structuring output directories
- Verifying data completeness
- Basic quality assessment
- Formatting considerations for the analysis phase

4 Part 2: Analysis and Visualization with Index-Visualizer

4.1 Introduction to Index-Visualizer

Building upon the data acquisition skills from the previous section, participants now advance to Index-Visualizer, which transforms raw climate data into standardized climate indices with sophisticated visualization capabilities. This tool addresses several key challenges in climate data analysis:

- Converting raw climate variables into scientifically meaningful indices
- Analyzing climate patterns across different temporal scales
- Comparing datasets from different sources
- Creating publication-quality visualizations
- Performing historical climate trend analysis

By providing a comprehensive framework for these tasks, Index-Visualizer enables participants to extract valuable insights from the climate data they acquired in the previous section.

4.2 Understanding Climate Indices

Before diving into the tool's functionality, participants will receive an overview of climate indices and their importance:

- **Definition:** Standardized metrics derived from climate variables that quantify specific climate characteristics
- **Scientific significance:** How indices provide more meaningful information than raw variables
- **Categories of indices:** Temperature-based, precipitation-based, and composite indices
- **Temporal considerations:** Annual, monthly, and event-based indices

This theoretical foundation prepares participants to make informed choices when configuring the Index-Visualizer for specific research questions.

4.3 Core Components and Architecture

Participants will explore the architecture of Index-Visualizer through its key components:

4.3.1 Analysis Engine

This component serves as the computational core of Index-Visualizer, responsible for:

- Calculating standardized climate indices from raw data
- Processing spatial and temporal dimensions
- Implementing scientific algorithms for index computation
- Optimizing performance for large datasets

Participants will examine how the Analysis Engine interfaces with Earth Engine to perform calculations efficiently, even for large regions or long time periods.

4.3.2 Data Manager

The Data Manager component provides:

- Access to multiple climate datasets (ERA5, PRISM, DAYMET)
- Standardized interfaces for different data sources
- Conversion factors for unit harmonization
- Metadata management for dataset properties

This section includes exercises on configuring dataset properties and understanding how the Data Manager integrates with the data acquired in Part 1.

4.3.3 Geometry Manager

Building on the area selection concepts from GeoClimate-Fetcher, the Geometry Manager offers:

- Multiple methods for defining analysis regions
- Interactive map-based drawing
- Shapefile import capabilities
- Coordinate-based boundary definition

Participants will practice defining analysis regions that match their study areas from Part 1, maintaining consistency across the workflow.

4.3.4 Visualization System

The visualization component provides sophisticated capabilities for:

- Creating interactive maps of climate indices
- Generating temporal trend plots
- Producing comparison visualizations
- Exporting publication-quality figures

Hands-on exercises will guide participants through creating multiple visualization types and understanding their appropriate applications.

4.4 Working with Climate Indices

Participants will explore the available climate indices in detail:

4.4.1 Temperature Indices

- **Annual maximum temperature:** The highest daily maximum temperature within a year
- **Annual minimum temperature:** The lowest daily minimum temperature within a year
- **Frost days:** Annual count of days with minimum temperature below 0°C
- **Summer days:** Annual count of days with maximum temperature above 25°C

Exercises will include calculating these indices for the study area selected in Part 1, examining their spatial patterns, and analyzing their trends over time.

4.4.2 Precipitation Indices

- **Annual total precipitation:** Total precipitation within a year
- **Annual maximum 1-day precipitation:** Maximum 1-day precipitation amount
- **Number of wet days:** Annual count of days with precipitation 1mm
- **Consecutive dry days:** Maximum number of consecutive days with precipitation ¡1mm

Participants will calculate these indices using the precipitation data downloaded in Part 1, visualize their spatial distribution, and examine their inter-annual variability.

4.5 Comparative Analysis Features

A distinctive feature of Index-Visualizer is its comparative analysis capability:

- Side-by-side analysis of different datasets
- Comparison of different time periods
- Contrasting different indices for the same region
- Statistical evaluation of differences

Hands-on exercises will guide participants through setting up comparison panels, transferring settings between panels, and interpreting comparative results.

4.6 Command-Line Interface

For participants interested in automation and scripting, this section covers the command-line interface:

- Basic CLI syntax and commands
- Automating index calculations
- Batch processing multiple regions or time periods
- Integrating CLI operations into larger workflows

Example commands and scripts will be provided for participants to adapt to their specific needs.

4.7 Hands-On Exercise: Climate Change Analysis

Participants will apply their knowledge in a comprehensive exercise:

1. Calculate maximum temperature and total precipitation indices for two time periods (1980-1990 and 2010-2020)
2. Compare the two periods using the comparison feature
3. Create visualizations showing the differences
4. Export the results for further analysis
5. Interpret the findings in the context of climate change

This exercise prepares participants for the validation phase in the next section by generating climate indices that will be evaluated against ground observations.

4.8 Exporting Results for Validation

As a transition to the next section, participants will learn how to properly export their analysis results:

- Exporting calculated indices in appropriate formats
- Preparing metadata for validation processes
- Structuring outputs for seamless integration with GeeData-GroundData-validator
- Documentation best practices for ensuring reproducibility

These skills ensure that the outputs from Index-Visualizer can be effectively utilized in the final phase of the workflow.

5 Part 3: Validation and Quality Assessment with GeeData-GroundData-validator

5.1 Introduction to GeeData-GroundData-validator

The final component of the integrated workflow addresses the critical need for validation—comparing gridded climate products against ground observations to assess their reliability. GeeData-GroundData-validator provides a comprehensive platform for:

- Retrieving ground station measurements from reliable sources
- Extracting gridded product values at station locations
- Performing statistical comparisons across multiple temporal scales
- Visualizing performance patterns through multiple representation methods
- Identifying optimal products for specific regions and applications

This tool completes the climate data workflow by enabling users to evaluate the quality of datasets and make informed decisions about which products are most suitable for their research needs.

5.2 User Interface and MVC Architecture

Participants will explore the Model-View-Controller architecture of GeeData-GroundData-validator:

- **User Interface Layer:** PyQt5-based graphical components
- **Controller Layer:** Application flow management and coordination
- **Business Logic Layer:** Core data fetching and analysis algorithms

- **Data Layer:** Storage and configuration management
- **Utility Layer:** Common functions used across the application

This architectural overview helps participants understand how the various components of the application interact and how they might extend or customize the tool for their specific needs.

5.3 Data Sources and Acquisition

A key component of validation is acquiring reliable ground data:

5.3.1 Ground Station Data: Meteostat

Participants will learn about:

- The Meteostat Python library and API
- Characteristics of station data (temporal coverage, variables, quality control)
- Filtering stations based on data completeness and quality
- Handling station metadata and location information

Hands-on exercises will guide participants through searching for stations, retrieving their metadata, and downloading precipitation data for their study area.

5.3.2 Geographic Selection Methods

Building on concepts from previous tools, GeeData-GroundData-validator offers three methods for defining study areas:

- US states selection (individual states or all US states)
- HUC watershed selection (based on Hydrologic Unit Codes)
- Custom polygon drawing on an interactive map

Participants will practice each method, understanding how the selection affects the stations included in the analysis.

5.3.3 Gridded Precipitation Products

This section covers the gridded products available for validation:

- ERA5: Global reanalysis at 11 km resolution
- DAYMET: North American dataset at 1 km resolution
- PRISM: Continental US dataset at 4 km resolution

- CHIRPS: Quasi-global rainfall dataset at 5.5 km resolution
- FLDAS: Land Data Assimilation System for drought monitoring
- GSMAP: Global Satellite Mapping of Precipitation
- GLDAS: Global Land Data Assimilation System (historical and current)

Participants will select products that match those analyzed in previous sections, ensuring continuity in the workshop workflow.

5.4 Statistical Analysis Framework

This section focuses on the statistical methods used for validation:

5.4.1 Performance Metrics

Participants will learn about the suite of metrics used to evaluate gridded products:

- **Coefficient of determination (R^2):** Measures variance explained
- **Root Mean Square Error (RMSE):** Quantifies average magnitude of errors
- **Bias:** Assesses systematic over/underestimation
- **Mean Absolute Error (MAE):** Measures average error magnitude
- **Nash-Sutcliffe Efficiency (NSE):** Evaluates predictive skill
- **Percent Bias (PBIAS):** Measures relative bias as a percentage

Exercises will include calculating these metrics for different products and interpreting their significance.

5.4.2 Temporal Scales and Aggregation

Validation can occur at multiple temporal scales:

- **Daily:** Direct comparison of daily values
- **Monthly:** Aggregated monthly totals or averages
- **Yearly:** Annual values for long-term assessment
- **Seasonal:** Performance during specific seasons

Participants will learn how to configure the analysis for different temporal scales and understand how performance can vary between them.

5.4.3 Statistical Filtering

To ensure robust analysis, the tool includes advanced filtering capabilities:

- Direction-aware filtering based on metric characteristics
- Configurable percentile thresholds
- Preservation of good performance metrics
- Impact of filtering on summary statistics

Hands-on exercises will demonstrate how filtering affects analysis results and visualization quality.

5.5 Visualization Generation

A key strength of GeeData-GroundData-validator is its comprehensive visualization capabilities:

5.5.1 Spatial Distribution Maps

These visualizations show performance metrics across space:

- Station-based maps with color-coded performance
- Regional patterns in product accuracy
- Identification of areas with consistent performance
- Influence of geography on product reliability

Participants will generate these maps for their study area and interpret the spatial patterns.

5.5.2 Time Series Comparisons

These visualizations focus on temporal performance:

- Overlay of ground and gridded time series
- Identification of timing errors in precipitation events
- Assessment of magnitude errors in precipitation intensity
- Seasonal patterns in agreement/disagreement

Exercises will include creating and interpreting time series for selected stations.

5.5.3 Box Plots and Distribution Visualizations

These tools summarize performance across stations:

- Distribution of performance metrics
- Identification of outlier stations
- Comparison of performance across products
- Statistical significance of differences

Participants will generate these visualizations and use them to compare the products analyzed in previous sections.

5.5.4 Radar Charts and Multi-dimensional Comparison

These advanced visualizations enable holistic comparison:

- Multi-metric comparison on a single chart
- Identification of product strengths and weaknesses
- Complementary performance across different metrics
- Overall ranking based on multiple criteria

This section culminates in creating comprehensive comparison visualizations that inform product selection.

5.6 Hands-On Exercise: Regional Validation

Participants will apply their knowledge in a comprehensive validation exercise:

1. Select the same study area used in previous sections
2. Configure the tool to analyze the gridded products from Parts 1 and 2
3. Retrieve ground station data for the region
4. Execute the full validation workflow
5. Generate comprehensive visualizations of product performance
6. Identify the best-performing products for the region

This exercise reinforces the integrated workflow by validating the same datasets that were acquired and analyzed in previous sections.

6 Integrated Case Study: From Acquisition to Validation

6.1 Case Study Overview

To consolidate learning across all three tools, participants will complete an integrated case study that demonstrates the full climate data workflow. This case study focuses on analyzing precipitation patterns in a selected watershed, addressing the following research questions:

- How has precipitation changed in the watershed over the past 20 years?
- Which precipitation products most accurately represent conditions in the region?
- What are the seasonal patterns and trends in precipitation?
- How reliable are different products for capturing extreme precipitation events?

This case study provides a practical context for applying all the skills learned throughout the workshop.

6.2 Step 1: Data Acquisition with GeoClimate-Fetcher

Participants will:

- Select a specific watershed using the drawing tool or HUC selection
- Configure time parameters for a 20-year period (2000-2020)
- Download precipitation data from multiple sources (ERA5, DAYMET, PRISM, etc.)
- Extract both time-series and gridded outputs for the region
- Organize the downloaded data for subsequent analysis

6.3 Step 2: Index Calculation with Index-Visualizer

Building on the acquired data, participants will:

- Load the precipitation data into Index-Visualizer
- Calculate multiple precipitation indices:
  - Annual total precipitation
  - Maximum 1-day precipitation
  - Number of wet days
  - Consecutive dry days
- Create visualizations showing spatial patterns and temporal trends

- Compare indices from different data sources
- Identify potential trends over the 20-year period
- Export the calculated indices and visualizations

6.4 Step 3: Validation with GeeData-GroundData-validator

For the final phase, participants will:

- Configure GeeData-GroundData-validator for the same watershed
- Retrieve ground station data for the region
- Extract gridded product values at station locations
- Calculate performance metrics at daily, monthly, and yearly scales
- Generate comprehensive visualizations of product performance
- Identify which products perform best for the region
- Create a summary of findings with supporting visualizations

6.5 Integration and Synthesis

The case study concludes with participants synthesizing their findings across all three tools:

- Connecting patterns observed in Index-Visualizer with validation results
- Using validation outcomes to select optimal datasets for future work
- Creating a presentation that documents the complete workflow
- Discussing how this approach could be applied to their own research

This comprehensive case study demonstrates how the three tools work together to create a powerful climate data processing pipeline, from initial acquisition through analysis to validation.

7 Advanced Topics and Customization

7.1 Extending the Tools

Each tool in the workshop can be extended or customized to meet specific research needs:

7.1.1 GeoClimate-Fetcher Extensions

- Adding custom datasets to the catalog
- Implementing new download formats
- Creating batch processing scripts
- Integrating with other Python libraries

7.1.2 Index-Visualizer Extensions

- Creating custom climate indices
- Implementing new visualization types
- Extending the command-line interface
- Adding new data sources

7.1.3 GeeData-GroundData-validator Extensions

- Incorporating additional gridded products
- Implementing new statistical metrics
- Creating custom visualization templates
- Extending geographic coverage beyond the US

7.2 Integration with Research Workflows

Participants will learn strategies for integrating these tools into their existing research workflows:

- Connecting outputs to hydrological models
- Incorporating results into GIS platforms
- Automating regular data updates
- Creating reproducible research pipelines

7.3 Performance Optimization

For participants working with large datasets or regions, this section covers:

- Strategies for efficient processing of high-resolution data
- Memory management techniques
- Parallel processing approaches
- Cloud-based computation options

8 Conclusion and Future Directions

8.1 Workshop Summary

This workshop has presented an integrated approach to climate data processing, addressing challenges at each stage of the workflow:

- **Data Acquisition:** GeoClimate-Fetcher provides intuitive access to diverse climate datasets from Google Earth Engine
- **Analysis and Visualization:** Index-Visualizer transforms raw data into meaningful climate indices with advanced visualization
- **Validation and Quality Assessment:** GeeData-GroundData-validator enables rigorous evaluation of gridded products against ground observations

Together, these tools form a coherent pipeline that enables researchers to work efficiently with climate data, regardless of their programming expertise.

8.2 Ongoing Development

All three tools are under active development, with planned enhancements including:

- **GeoClimate-Fetcher:** Expanded dataset catalog, improved handling of very large regions, and enhanced metadata management
- **Index-Visualizer:** Additional climate indices, more visualization options, and improved comparison capabilities
- **GeeData-GroundData-validator:** Extended geographic coverage, additional ground data sources, and enhanced statistical methods

8.3 Community Engagement

The tools presented in this workshop are open-source projects that welcome community contributions:

- GitHub repositories with issue tracking and feature requests
- Documentation and tutorials for new users
- Opportunities for code contributions and extensions
- User forums for questions and discussion

Participants are encouraged to engage with these projects, share their experiences, and contribute to their ongoing development.

8.4 Next Steps for Participants

To continue building on the skills learned in this workshop, participants are encouraged to:

- Apply the tools to their own research questions
- Share feedback on their experiences
- Explore the advanced features not covered in detail during the workshop
- Consider contributing to the open-source development of these tools

9 Acknowledgments

The development of these tools and this workshop has been supported by:

- Department of Civil and Environmental Engineering, Jackson State University
- U.S. Army Corps of Engineers, Engineer Research and Development Center
- Google Earth Engine Research Awards Program

Special thanks to the open-source communities behind the libraries that power these tools, including GeoPandas, Matplotlib, PyQt5, and the Earth Engine Python API.

10 References

References

[1] Bhattarai, S. (2023). GeoClimate-Fetcher: An open-source Python package for downloading Google Earth Engine (GEE) climate data for user-defined study areas. GitHub Repository. https://github.com/Saurav-JSU/GeoClimate-Fetcher

[2] Bhattarai, S. (2023). Index-Visualizer: A comprehensive Python-based tool for analyzing climate data using Google Earth Engine. GitHub Repository. https://github.com/Saurav-JSU/Index-Visualizer

[3] Bhattarai, S. (2023). GeeData-GroundData-validator: A PyQt5 application for retrieving, analyzing, and visualizing precipitation data from multiple sources. GitHub Repository. https://github.com/Saurav-JSU/GeeData-GroundData-validator

[4] Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment, 202, 18-27.

[5] Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... Thépaut, J. N. (2020). The ERA5 global reanalysis. Quarterly Journal of the Royal Meteorological Society, 146(730), 1999-2049.

[6] Thornton, P. E., Thornton, M. M., Mayer, B. W., Wilhelmi, N., Wei, Y., Devarakonda, R., Cook, R. B. (2014). Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 2. ORNL DAAC.

[7] Daly, C., Halbleib, M., Smith, J. I., Gibson, W. P., Doggett, M. K., Taylor, G. H., ... Pasteris, P. P. (2008). Physiographically sensitive mapping of climatological temperature and precipitation across the conterminous United States. International Journal of Climatology, 28(15), 2031-2064.

[8] Funk, C., Peterson, P., Landsfeld, M., Pedreros, D., Verdin, J., Shukla, S., ... Michaelsen, J. (2015). The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. Scientific Data, 2(1), 1-21.

[9] Alexander, L. V., Zhang, X., Peterson, T. C., Caesar, J., Gleason, B., Klein Tank, A. M. G., ... Vazquez-Aguirre, J. L. (2006). Global observed changes in daily climate extremes of temperature and precipitation. Journal of Geophysical Research: Atmospheres, 111(D5).

A Prerequisites Checklist

Required Software:

- ☐ Python 3.8-3.12 (not 3.13)
- ☐ Git version control system
- ☐ Visual Studio Code (recommended)
- ☐ Conda or pip package management

Accounts:

- ☐ Google Earth Engine account (approved)
- ☐ Google Cloud Project with Earth Engine API enabled
- ☐ GitHub account (for accessing repositories)

System Requirements:

- ☐ 8 GB RAM (16 GB recommended)
- ☐ 10 GB free disk space
- ☐ Administrator privileges for installation
- ☐ Stable internet connection

B Troubleshooting Guide

GeoClimate-Fetcher Installation Issues:

Issue: Permission error during installation **Solution:** Run the install command again or use `--user` flag with pip

Issue: Missing dependencies
Solution: Install dependencies individually: `pip install earthengine-api geemap`

Issue: Earth Engine authentication failure
Solution: Verify account is approved and run `earthengine authenticate` separately

Index-Visualizer Issues:

Issue: Conda environment creation fails
Solution: Use pip with requirements.txt instead of conda

Issue: Jupyter notebook does not display widgets
Solution: Install and enable required extensions: `jupyter nbextension enable --py widget`

Issue: Map visualization does not appear
Solution: Ensure ipyleaflet is properly installed and enabled

GeeData-GroundData-validator Issues:

Issue: PyQt5 installation problems
Solution: On Linux, install system packages: `sudo apt-get install python3-pyqt5`

Issue: GDAL/GeoPandas installation errors
Solution: Use conda to install these packages: `conda install -c conda-forge geopandas`

Issue: Application crashes during map interaction
Solution: Verify Earth Engine authentication and project ID configuration

C Workshop Datasets

Example Study Areas:

Mississippi River Basin (HUC 08)

California Central Valley

Florida Everglades

Colorado River Watershed

Recommended Datasets:

ERA5 Daily Aggregates

PRISM AN81d

DAYMET V4

CHIRPS Daily

Time Periods for Exercises:

Recent past: 2010-2020

Mid-term comparison: 1990-2000 vs. 2010-2020

Long-term trends: 1980-2020