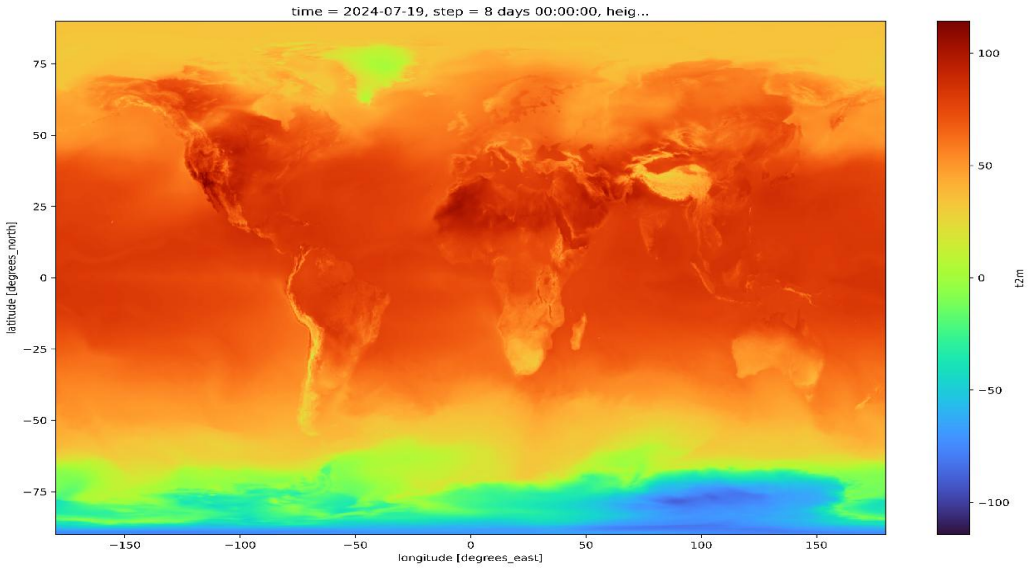
A Python Workshop to Efficiently Access, Deliver, Visualize, and Analyze CMIP6-like Big Climate Data in the Cloud

Instructors

Ryan Paul Lafler
Founder, CEO, Chief Data Scientist, Lead Consultant and Adjunct Faculty
Premier Analytics Consulting, LLC and San Diego State University
E-mail: rplafler@premier-analytics.com
LinkedIn: www.LinkedIn.com/in/RyanPaulLafler
Website: www.Premier-Analytics.com

Samuel Shen
Distinguished Professor; Faculty and Co-founder of the Big Data Analytics MS Program
San Diego State University
E-mail: sshen@sdsu.edu
Website: Shen.SDSU.edu/researchGroup.html

Mitch Goldberg
Distinguished Research Scientist
City College of New York / NOAA-CESSRST Program
E-mail: mgoldberg1@ccny.cuny.edu
LinkedIn: www.LinkedIn.com/in/Mitch-Goldberg
Website: www.CESSRST.org/team/Mitch-Goldberg/



Workshop Description

Join us for an immersive, full-day workshop designed to equip attendees with practical skills in accessing, processing, analyzing, and visualizing large-scale climate datasets stored in cloud repositories. This full-day Python

Copyright © 2024-2025 by Premier Analytics Consulting, LLC
Developed by Ryan Paul Lafler

programming workshop provides attendees with the applied experience necessary to programmatically access and efficiently process, analyze, and visualize massive climate datasets inside of cloud storage providers.

Attendees will learn to navigate and leverage CMIP6-like climate data repositories stored in Google Cloud Storage and Amazon S3. We'll develop efficient cloud pipelines using Python that process, load, and prepare climate data from the ECMWF (European Centre for Medium-Range Weather Forecasts) and NOAA-RTMA (Real-Time Mesoscale Analysis) repositories to generate stunning visualizations and conduct data-driven analysis.

Through hands-on training, attendees will leverage Python libraries capable of processing large-scale gridded climate data from cloud storage repositories. The skills gained from this workshop will allow attendees to filter, extract, slice, manipulate, analyze, and visualize different climate datasets, climate attributes, and temporal resolutions from any CMIP6-like climate data repository.

At the end of this Python workshop, attendees will be able to:

- Navigate high-resolution CMIP6-like cloud repositories with ease using Python,
- Build high-speed climate data pipelines that connect to cloud-object storage providers,
- Analyze ECMWF and NOAA-RTMA high resolution climate datasets,
- Access, slice, query, and generate time series from gridded data using Python's *Xarray* library,
- Generate a virtual ZARR store from raw gridded data for quick access in the cloud,
- Efficiently manage and process big "out-of-memory" climate records using Python libraries,
- Create stunning static, interactive, and animated visualizations of big climate data.

Workshop Target Audience

This application-oriented Python programming workshop is open to all researchers, climate scientists, students, educators, data scientists, programmers, machine learning engineers, and GIS analysts interested in learning, enhancing, and/or refining their Python programming skills towards large-scale, data-driven climate science.

Workshop Agenda

This full-day Python workshop will consist of the following topics.

Proposed Time (Eastern)	Topic #	Topic Title
8:00 AM – 8:30 AM (30 min)	**1**	Introductions, Workshop Objectives, and Python Libraries Setup
8:30 AM – 9:00 AM (30 min)	**2**	Spatiotemporal Data Characteristics, Properties, Filetypes, and Formats
9:00 AM – 9:45 AM (45 min)	**3**	Python Fundamentals for Climate Data in the Cloud: Libraries, Packages, and Concepts
9:45 AM – 10:15 AM (30 min)		**Coffee Break, Questions, and Breakout Discussions**
10:15 AM – 11:15 AM (60 min)	**4**	Navigate, Access, and Explore Historical High Resolution ECMWF Datasets in GCS using Python Pipelines

Copyright © 2024-2025 by Premier Analytics Consulting, LLC
Developed by Ryan Paul Lafler

11:15 AM – 12:15 PM (60 min)	**5**	Process, Filter, Extract, Visualize, and Generate Time Series from ECMWF Data using the Xarray Library
12:15 PM – 1:15 PM (60 min)		**Lunch Break, Questions, and Breakout Discussions**
1:15 PM – 2:15 PM (60 min)	**6**	Efficiently Analyze and Visualize Multi-Dimensional ECMWF Air Temperature Models at Different Air Pressure Levels
2:15 PM – 3:15 PM (60 min)	**7**	Navigate, Access, and Explore Hourly RTMA Climate Data in Amazon S3 using Python Pipelines
3:15 PM – 3:45 PM (30 min)		**Coffee Break, Questions, and Breakout Discussions**
3:45 PM – 4:45 PM (60 min)	**8**	Generate a Virtual ZARR Store from Raw GRIB Files for Quick Cloud Access using Hourly RTMA Data
4:45 PM – 5:00 PM (15 min)		**Workshop Recap, Questions, and Final Discussions**

Detailed Agenda

2) Spatiotemporal Data Characteristics, Properties, Filetypes, and Formats

- Characteristics of spatiotemporal data → spatial resolution, temporal resolution, coordinate reference systems, projections
- Differences between gridded (raster) data and vector (station, boundary, polygons) data
    - Examples of raster file formats and their properties:
        - NetCDF, GRIB, GRIB2, Zarr archives, DEM, COG, GeoTIFF, PNG, JPEG
        - Reasons for choosing one file type over another, scalability, and import speeds
    - Examples of vector file formats and their properties:
        - CSV, Apache Parquet, GeoJSON, ESRI Shapefile, Geodatabase
- Virtual ZARR Stores → accessing gridded data in cloud environments quickly and efficiently

3) Python Fundamentals for Climate Data in the Cloud: Libraries, Packages, and Concepts

- Python programming objects and methods → better programming for customizable, data-driven analysis
- Libraries for traditional data processing and spatial data processing
- Libraries that extend the capabilities of *Pandas* and *NumPy* → process and work with big datasets
    - Efficiently manage “out-of-memory” climate datasets using *Dask*
    - Process gridded climate datasets using *Xarray*
    - Connect to GCS and Amazon S3 using the *fsspec* library
- Libraries that retrieve data from cloud-object storage providers
- Libraries for static, interactive, and animated visualizations of climate data

Copyright © 2024-2025 by Premier Analytics Consulting, LLC
Developed by Ryan Paul Lafler

4) Navigate, Access, and Explore Historical High Resolution ECMWF Datasets in GCS using Python Pipelines

- Explore the structure, format, and filetypes for CMIP6 data in Google Cloud Storage (GCS)
- Engineer Python pipelines that access and deliver specified data from GCS into the Python session
- Incorporate Python libraries to efficiently handle and process multidimensional climate data
- Develop, test, and deploy pipelines to query different climate attributes
- Examine the performance of the cloud pipeline → investigate data chunking and loading strategies

5) Process, Filter, Extract, Visualize, and Generate Time Series from ECMWF Data using the Xarray Library

- Investigate the properties, attributes, and metadata associated with ECMWF high resolution records
- Specify keys to filter for specific climate attributes
- Filter and aggregate ECMWF records over temporal and spatial dimensions
- Generate static and animated visualizations of daily ECMWF temperature records with *Matplotlib*
- Generate time series for specific locations; create comparative time series

6) Efficiently Analyze and Visualize Multi-Dimensional ECMWF Air Temperature Models at Different Air Pressure Levels

- Investigate efficient chunking strategies to efficiently handle large multi-dimensional arrays
- Perform multi-dimensional slicing to extract specific arrays and points by air pressure levels
- Generate comparative time series to examine air temperature at different pressure levels
- Create animated visualizations that show changes in air temperature by pressure levels
- Calculate and compare differences in air temperature by pressure levels

7) Navigate, Access, and Explore Hourly RTMA Climate Data in Amazon S3 using Python Pipelines

- Explore the structure, format, and filetypes for NOAA-RTMA data in Amazon S3
- Engineer Python pipelines that access and deliver specified data from Amazon S3 into the Python session
- Leverage the *fsspec*, *Xarray*, and *Dask* libraries to efficiently process RTMA data from the cloud
- Examine the performance of the cloud pipeline → investigate data chunking and loading strategies
- Query, Retrieve, and Visualize hourly RTMA temperature datasets with *Xarray*

8) Generate a Virtual ZARR Store from Raw GRIB Files for Quick Cloud Access using Hourly RTMA Data

- Understand the purpose of creating virtual ZARR stores from NetCDF, GRIB, and ZARR archives
- Explore the *VirtualiZarr* and *Kerchunk* libraries to generate virtual ZARR stores
- Create a virtual ZARR store for a single hourly RTMA file merging all climate variables
- Create a virtual ZARR store for 2-days' worth of hourly RTMA data
- Investigate the outputted JSON metadata file structure, components, and purpose
- Save, export, and import the virtual ZARR JSON file into Python using the *Xarray* library

Copyright © 2024-2025 by Premier Analytics Consulting, LLC
Developed by Ryan Paul Lafler