SOFTWARE COMPONENTS

Our software consists of two main components: the Data Wrangler and the Dashboard Generator, which are described in more detail below. The Data Wrangler generates processed data files that can be saved externally and then loaded by the Dashboard Generator when the user launches the dashboard application. The Dashboard Generator creates an interactive dashboard that provides climate data for a specific location. The overall interaction between these two components is shown schematically in Figure 1.

Having the processed data already saved at the point of user launch will enable the dashboard application to function quickly (i.e. not need to reprocess the raw data each time). While it will not be necessary to run the Data Wrangler often, it is included in the software package to document how the raw data was processed and to enable users to update dashboards with new climate model and observational data in the future (e.g. additional climate model data is expected to become available in the next several months).

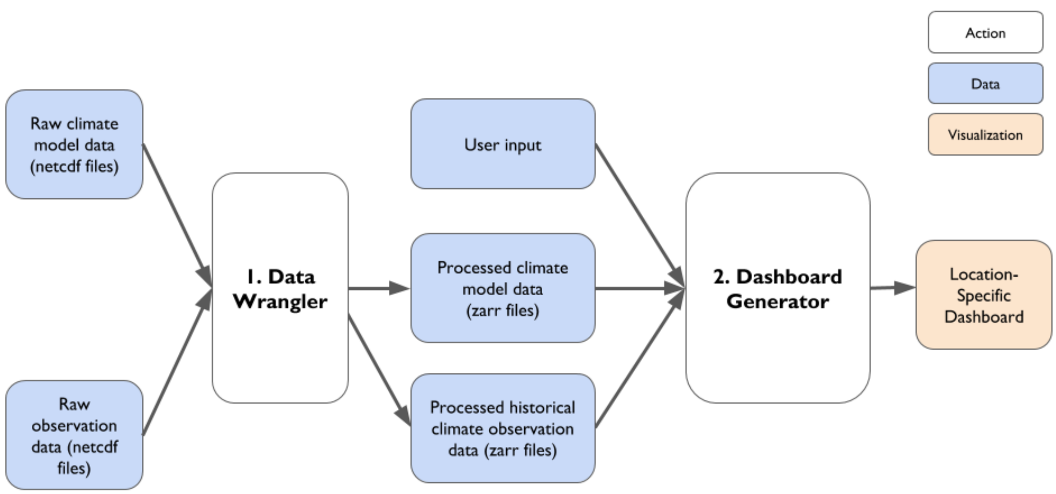

Figure 1: High-level schematic of interaction between two main software components.

1. Data Wrangler

What it does: The Data Wrangler processes raw modeled and observational climate data to produce consistently formatted files that can be used to generate visualizations. This model component is necessary for a variety of reasons:

- Datasets containing climate model output are formatted and structured differently than datasets of historical observations; in order to compare climate models with observations, it is necessary to process climate model output and historical observations into a consistent format.
- While at a high level all the climate model output datasets have the same geospatial coordinates (latitude, longitude), all datasets are on different geospatial grids (i.e. some are higher resolution than others) and need to be processed (i.e. regridded to consistent grid) to be able to make calculations across models.
- The observational and modeled datasets have differently formatted time (i.e. different data types) that need to be made consistent in order to visualize data across different datasets.
- The raw climate model output datasets are too large for quick and responsive use in an interactive dashboard, and contain more detail than is necessary for our target use cases (e.g. many modeling centers provide model output for multiple ensemble members[1]). Because of this, it is necessary to compress the large datasets into summary statistics datasets that are smaller in size but still contain information at a level of detail that can be useful to our target users.

Various subcomponents (A-D) interact as part of the Data Wrangler, as illustrated schematically in Figure 2.

Required inputs (see Data Description for more detail about the raw data inputs):

- Raw climate model data (netcdf files[2]):
    - 1 variable: average monthly temperature[3]
    - 19 climate models
    - 5 scenarios (historical, SSP1, SSP2, SSP3, and SSP5)
    - 1 netcdf file per variable, model, scenario, and ensemble member
- Raw observation data (netcdf files)
    - 1 netcdf file per variable

[1] Some modeling centers run different ensemble members for a given scenario, which consists of running the scenario multiple times with very slightly different initial conditions, which are generated by perturbing initial conditions at the level of round-off error. These ensemble members help quantify the internal variability of the climate system, but provide a level of detail beyond the scope of this project.

[2] Netcdf is a file format for array-oriented scientific data, and it is commonly used for climate data. It is the original file type for the raw climate model output and the historical observation datasets.

[3] Note that we have structured our code to be easily extendable to incorporate additional variables (e.g. average precipitation, maximum temperature) in the future.

Outputs provided:

- Processed observation data formatted for use in dashboard generator (`historical_obs.zarr`[4])
- Processed climate model data for use in dashboard generator (1 zarr file per variable and scenario, with file format `modelData_VARIABLE_SCENARIO.zarr`)

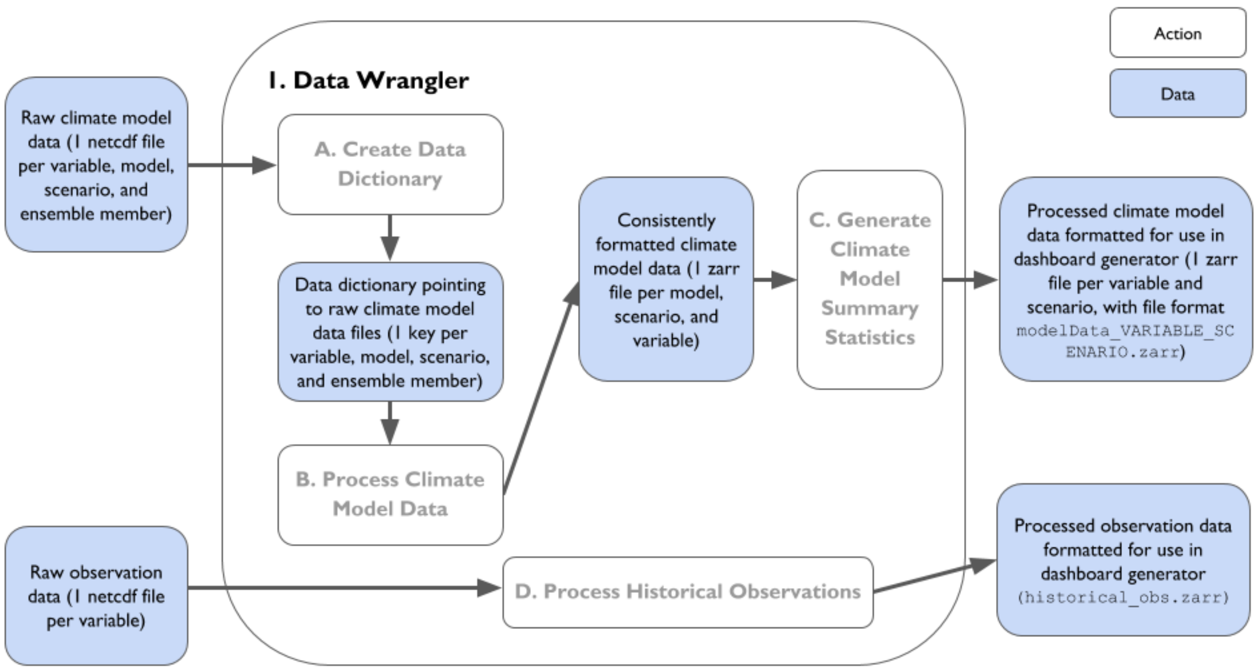

Figure 2: Schematic of interaction between Data Wrangler software subcomponents.

Subcomponents:

A. Create Data Dictionary

B. Process Climate Model Data

C. Generate Climate Model Summary Statistics

D. Process Historical Observations

[4] Zarr is a file format for array-oriented scientific data. Zarr is a newer file type than netcdf, but increasingly popular in the climate sciences. Zarr files are a more efficient and direct file type for storage on the cloud, which is where we ultimately aim to host our processed data files. In Zarr, it is also possible to save data in "chunks" of individual arrays, which enable us to avoid loading all global data to memory. Instead just loading the grid cells surrounding the latitude and longitude the user selects.

2. Dashboard Generator

What it does: The Dashboard Generator creates and interactively updates a dashboard with location-specific (i.e. for a user-provided geographic location) modeled and observed climate data. Various subcomponents (A-F) acquire user input, generate visualizations, and update visualizations based on user interaction, as illustrated schematically in Figure 3.

Required inputs:

- Climate data
  - Processed observation data formatted for use in dashboard generator (`historical_obs.zarr`)
  - Processed climate model data for use in dashboard generator (1 zarr file per variable and scenario, with file format `modelData_VARIABLE_SCENARIO.zarr`)
- User input: geospatial selections
  - Latitude
  - Longitude
- User input: categorical selections
  - Scenario (historical, SSP1, SSP2, SSP3, SSP)[5]
  - Whether or not to show annual average or monthly data

Outputs provided: Dashboard with the following modules

- Time Series Plot: Scenario Uncertainty (interactive)
- Time Series Plot: Model Uncertainty (interactive)
- Settings panel documenting user selections (interactive)
- Abbreviated documentation (static)

[5] See Data Description or this article for a more detailed explanation of shared socioeconomic pathways (SSPs).

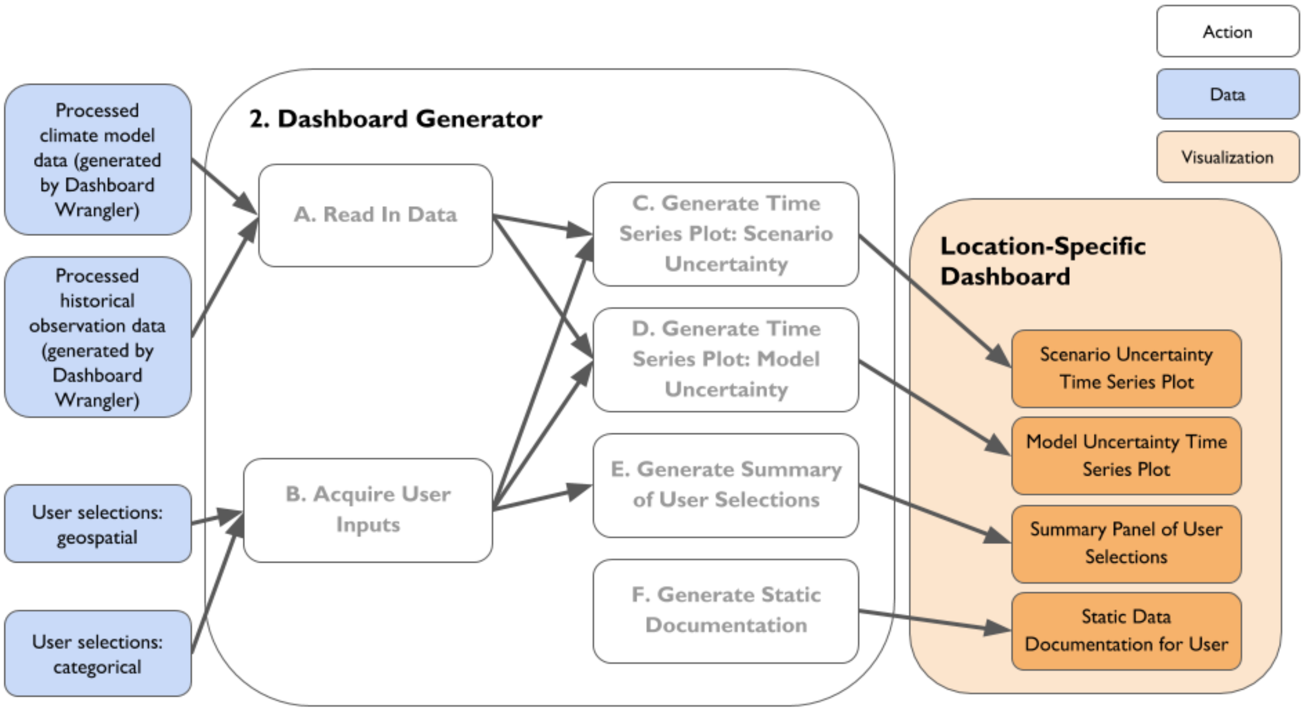

Figure 3: Schematic of interaction between Dashboard Generator software subcomponents.

Subcomponents:

A. Read In Data

Subcomponent A reads processed data output from the Data Wrangler (.zarr file) into an xarray format which is a multidimensional data array based on Numpy and Pandas packages.

B. Acquire User Inputs

Subcomponent 2B (1) translates user geospatial input into latitude and longitude coordinates to update the dashboard visualizations and (2) reads in user categorical selections.

Required inputs:

- One of the following:
  - User selection of city from drop-down menu
  - User input of specific latitude and longitude on slider widgets
- Binary selection: annual mean (indicating whether or not to average annual mean of variable or to display monthly values)
- Scenario selection

Outputs provided:

- Latitude
- Longitude
- Annual mean flag
- Scenario index

C. Generate Time Series Plot: Scenario Uncertainty (interactive)

Subcomponent C uses the geospatial output (latitude and longitude) from subcomponent B to subset the data read in from subcomponent A, and generate a time series plot of Scenario Uncertainty which is averaged (or not) according to the annual mean flag from subcomponent B.

D. Generate Time Series Plot: Model Uncertainty (interactive)

Subcomponent D uses the geospatial output (latitude and longitude) and scenario index from subcomponent B to subset the data read in from subcomponent A, and generate a time series plot of Model Uncertainty which is averaged (or not) according to the annual mean flag from subcomponent B.

E. Generate Summary of User Selections (interactive)

Subcomponent E uses the user selections indicated by subcomponent B to show update a panel summarizing user selections (e.g. indicating which location was selected).

F. Generate Static Documentation

Subcomponent F generates static formatted text that briefly document the data used by the dashboard and describe what the dashboard is showing.

3. Use Case Interactions

Users only interact with one component which is the Dashboard Generator. As stated above, the Data Wrangler component operates at the backend. As a result, use case interactions that are stated below are only explained with the subcomponents of the Dashboard Generator.

Use Case 1

Objective: The user wants to create a climate dashboard for her city to understand climate models and uncertainty.

Interactions:

Subcomponent 2B acquires user inputs for geospatial and categorial selections. For example, annual average temperature of Seattle. After the user entered their desired location and annual mean selection, 2A will read processed data output from the Data Wrangler output and create a subset including only values for the specified location. Using this subset, subcomponents 2C-2D will generate a dashboard that consists of two interactive time series plots for model and scenario uncertainty and 2E-2F will document the user selections and display detailed information about visualizations in a text box.

Use Case 2

Objective: The user wants to understand how the full range of uncertainty across models changes in a high-emissions scenario compared to a low-emissions scenario.

The user selects their location (for example: Seattle, USA) and selects a high-emissions scenario (SSP5) and clicks “Refresh plots.” Subcomponent 2B acquires these geographic and categorical user inputs and subcomponent 2A reads the model and observational data output by the Data Wrangler to create a subset based on the user selections. This subset will be used by 2C-2F to update the visualizations. The user can then change the scenario selection to a low-emissions scenario (SSP1) and click “Refresh plots.” The visualizations will update as described above, and the user can see how the range of modeled uncertainty changes.