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. 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. The overall interaction between these two components is shown schematically in Figure 1.

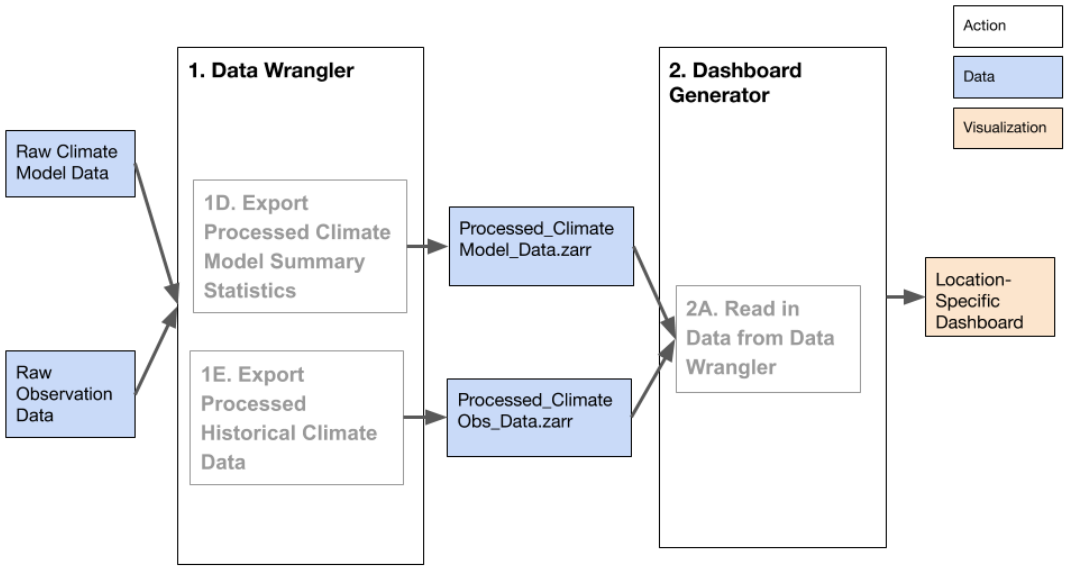

Figure 1: High-level schematic of interaction between two main software components.

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:

- Necessary to keep observations and model data consistent.
- The raw model data contains too much detail and is too big for quick and responsive use in a dashboard - we really just want to compress to some summary statistics.
- 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
- Need to average across ensemble members

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

Required inputs:

- Raw climate model data (netcdf files[1])
    - 3 variables: average, minimum and maximum monthly temperature

1 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.

- X models
- 5 scenarios (historical, SSP126, etc.)
- 1 netcdf file per variable, model, scenario, and ensemble member[2]

- Raw observation data (netcdf files)
  - 1 netcdf file per variable.

Outputs provided:

- Processed_Climate_Obs_Data.zarr[3]
- Processed_Climate_Model_Data.zarr

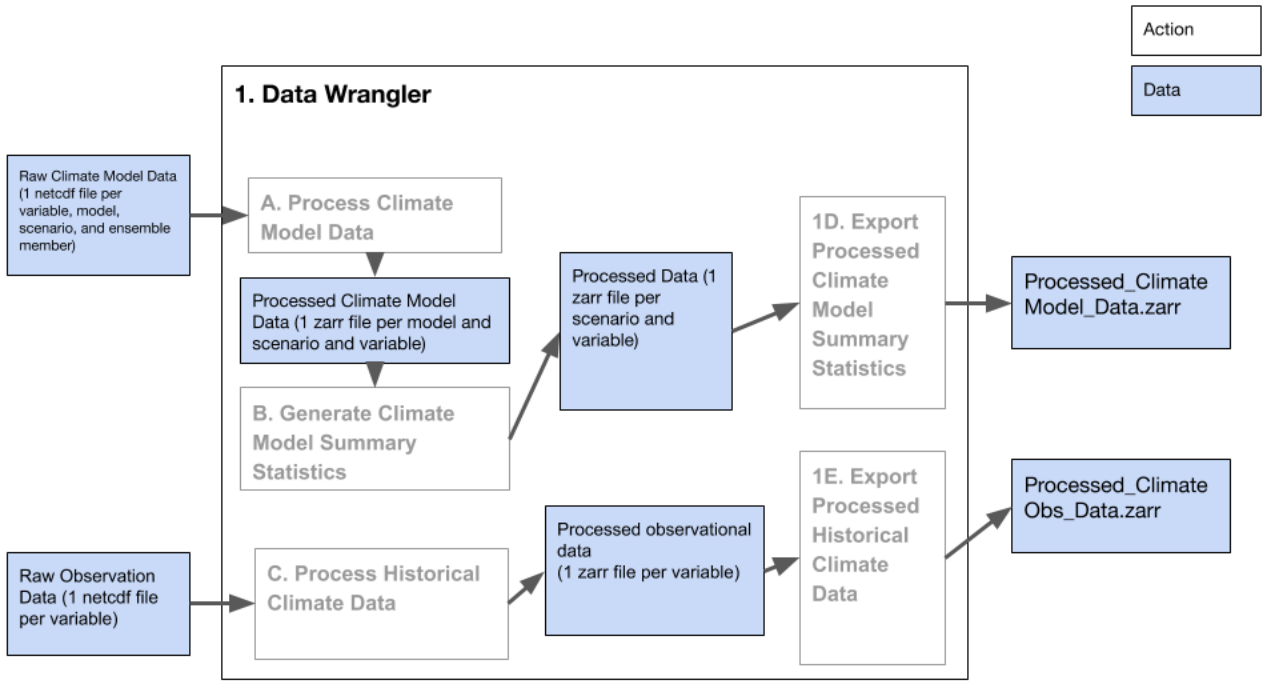

Figure 2: Schematic of interaction between Data Wrangler software subcomponents.

2 Climate model ensembles, such as the Coupled Model Intercomparison Project phase 6 (CMIP6), are used to characterize broadscale ranges of projected regional climate change and their impacts..

3 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 intermediate processed data files Processed_Climate_Model_Data and Processed_Climate_Obs_Data. 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.

A. Process Climate Model Data

B. Generate Climate Model Summary Statistics

C. Process Historical Climate Data

D. Export Processed Climate Model Summary Statistics

E. Export Processed Historical Climate Data

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-K) acquire user input, generate visualizations, and update visualizations based on user interaction, as illustrated schematically in Figure 3.

Required inputs:

- Processed_Climate_Obs_Data.zarr
- Processed_Climate_Model_Data.zarr
- User input: geospatial selections
    - Latitude
    - Longitude
- User input: categorical selections
    - Variable (average monthly temperature, minimum monthly temperature, maximum monthly temperature)
    - Scenario (historical, SSP126, SSP245, SSP370, SSP585)[4]

[4] **SSP585:** highest emission scenario. **SSP370:** medium to high forcing pathway, takes into account the land use changes and high emissions (esp. of SO2) **SSP245:** represents

- Time Format (Annual Average, May Average, Summer Average etc.)

Outputs provided: Dashboard with the following modules

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

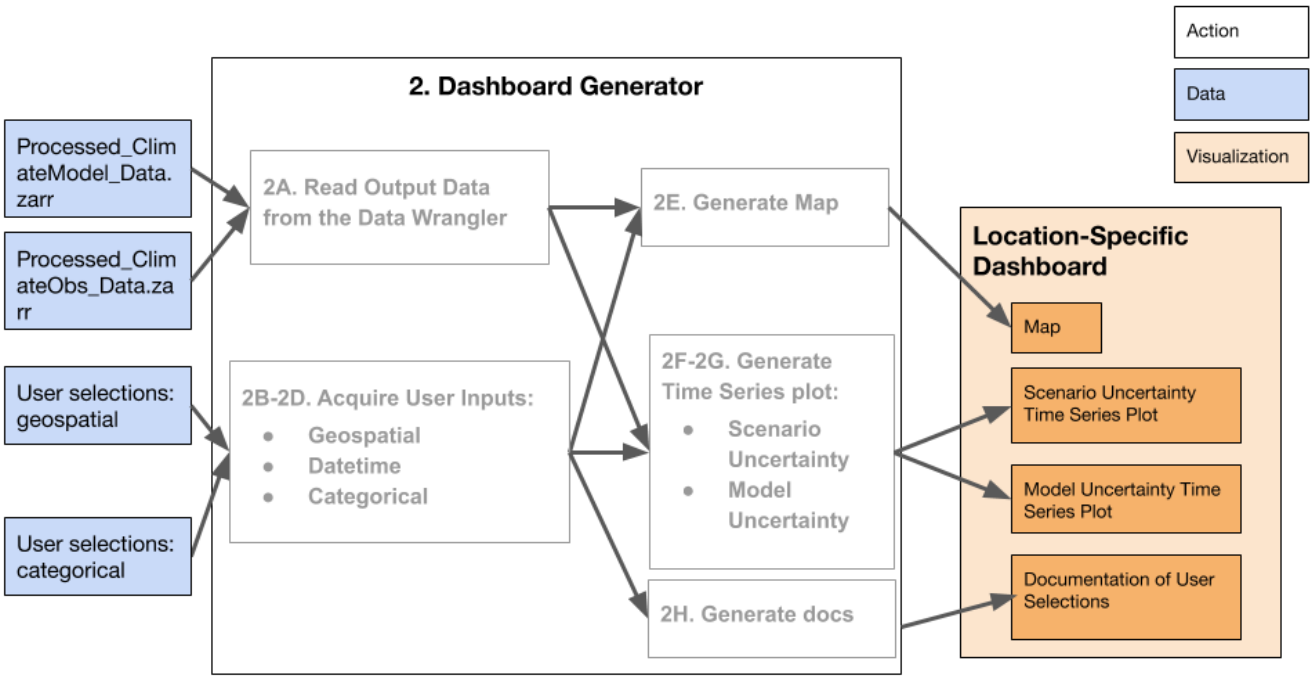

Figure 3: Schematic of interaction between Dashboard Generator software subcomponents.

regional downscaling, medium forcing **SSP126:** low emissions, future goals. *See here for a more detailed explanation.*

A. Read Output Data from the Data Wrangler

Component 2A reads processed data output from the Data Wrangler (.zarr file) into an Xarray format which is a multidimensional (ND) data arrays based on Numpy and Pandas packages.

B. Acquire User Input: Geospatial

Component 2B translates user geospatial input into latitude and longitude coordinates to update the dashboard visualizations.

Required inputs:

- One of the following:
  - User click on map
  - User selection of city from drop-down menu
  - User input of specific latitude and longitude on slider widgets

Outputs provided:

- Latitude
- Longitude

C. Acquire User Input: Datetime

Component 2C allows user to select a timespan from the drop-down menu Output: Datetime format and time interval.

D. Acquire User Input: Categorical

Component 2D allows user to change one of the 4 different scenarios from drop-down menu. Output: historical, SSP126, SSP245, SSP370, SSP585

E. Generate Map (interactive)

Component 2E uses the geospatial output from 2B and generate a world map with a pinpoint on the user-selected location.

F. Generate Time Series Plot: Scenario Uncertainty (inter.)

Component 2F uses the categorical output from 2D and generate a time series plot of Scenario Uncertainty.

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

Component 2G uses the categorical output from 2D and generate a time series plot of Scenario Uncertainty.

H. Generate Documentation Panel (static)

Component 2H uses the outputs from 2B-2D and generate a text box including the user-selections and detailed information about the visualizations.

3. Use Case Interactions

Users only interact with one component which is the Dashboard Generator. As stated above, Data Wrangler component operates at 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:

Subcomponents 2B and 2C acquire user inputs for geospatial and time-series selections. For example, annual average temperature of Seattle from 1996 to 2010. After the user entered their desired location and timespan, 2A will read processed data output from the Data Wrangler and create a subset including only data variables for the specified location with the specified timespan. Using this subset, subcomponents 2E-2G will generate a dashboard consists of two interactive time series plots for **default** model and scenario uncertainty, a map of the entered location presenting the difference between scenario and historical data and 2H will print out the user selections and detailed information about visualizations in a text box.

Use Case 2

Objective: The user wants to understand the changes in climate models and uncertainty when there is a change in scenario.

Subcomponent 2D acquire categorical user inputs for different climate models and scenarios. After the user entered their desired scenario, 2A will read relevant model data output from the Data Wrangler and create a subset for the same variables indicated in the subcomponents 2B and 2C. This subset will be used by 2E-2G to update the visualizations and 2H will print out the new information.

4. Preliminary plan

See Tasks spreadsheet

Task	Project Phase	Point Person	Priority	Helpers	Status
Create unit tests for raw observation data that we read in	1 - Data Wrangler	Ihsan	High		Not started
Subcomponent A: Process climate model data to have same lat/lon/time dimensions	1 - Data Wrangler	Claire	High	Jacqueline	In progress
Subcomponent B: Create summary statistics for climate model data (i.e. mean, min, max, stddev)	1 - Data Wrangler	Jacqueline	High	Claire	Not started
Pick data sets	1 - Data Wrangler	All	High		Done
Make dummy data for starting development of Dashboard Generator	1 - Data Wrangler	Claire	High	Jacqueline	Done
Design mockup app, specifying points for user input	2 - Design app	Claire	High	Sami, Jacqueline	Done
Create unit tests for processed climate model and observation data that we will load into app	3 - Dashboard Generator	Ihsan	High		Not started
Decide on timeline for different parts of analysis	Admin		High		Not started
Choose library for technology review	Documentation	Sami	High		Done
Make technology review presentation	Documentation	All	High	All	Done
Make Github README.md	Documentation	Sami	High		Done
Define the Problem	Documentation	Lauren	High		Done
Write the Component Specification	Documentation	Claire	High		Done
Write the Functional Specification	Documentation	Lauren	High	Jacqueline	Done
Make Github repository	Documentation	Claire	High		Done
Load zarr files onto server/cloud	4 - Bring app online		Low		Not started
Come up with name	Admin	Jacqueline	Low	All	Not started
Unit tests for subcomponent E	Testing	Ihsan	Low		In progress
Talk to Ryan Abernathy about data structure for	1 - Data Wrangler	Claire	Medium		Done

what is passed to app					
Create unit tests for raw model data that we read in	1 - Data Wrangler	Ihsan	Medium	Claire	Not started
Subcomponent C: Process observations to have consistent lat/lon/time dimensions as climate model data	1 - Data Wrangler	Jacqueline	Medium	Claire	Not started
Subcomponents D and E: Save processed data as zarr files (chunked by lat/lon)	1 - Data Wrangler	Claire	Medium	Jacqueline	Not started
Describe data sets	Documentation		Medium		Not started
Unit tests for subcomponent A	Testing	Ihsan	Medium		Not started
Unit tests for subcomponent B	Testing	Ihsan	Medium		Not started
Unit tests for subcomponent C	Testing	Ihsan	Medium		Not started
Unit tests for subcomponent D	Testing	Ihsan	Medium		Not started