Last updated: December 2025

# Git Documentation & Organization

**Goal:** The goal of the document is to provide information on the files available in Github, as well as those we could not make publicly available, so that a user could understand what information they would need to compile to run the model or apply it elsewhere.

**Notes:** If a name of a file is given, it is included in the Github repository. If the name is not given and the contents/information from it are included, then it is not publicly available.

## Subfolder: data

### costs

- File: Csv file with cost values for deployment order in linear programming
  - Columns: Category (name of source), Cost (values can be negative or positive)

### data_system

- File: Csv file with bathymetry information for Loch Lomond
  - Columns: Index, Stage (ft NCDD), Stage (ft NAVD88), Storage (acre-ft), Storage (Mgal), Area (ac)

### dcc_data

- File: Csv file with DCC model coefficients
  - Name: DCCcoefs.csv
- File: Csv file with single-family households data for use in simulation model
  - Columns: account, pipe size (in), pool, bathrooms, main area (sqft), tax value, census tract/block group, map_inc_1 … map_inc_10, mean residuals (real)

### Input_climate_stress_test

- Subfolder: FLOW
  - Files: Csv files for each stochastic simulation
    - Columns: Date; Streams (raw): Bigtrees, Laguna, Liddel, Majors, Newell, Taitside; Streams (Na): Bigtrees, Laguna, Liddel, Majors, Taitside; is it a critically dry year? (binary); first flush (binary)
- Subfolder: WEATHER
  - Files: Csv files for each stochastic simulation
    - Columns: Date, Precipitation, Evaporation, Turbidity (San Lorenzo River), Turbidity (North Coast), Turbidity (North Coast Majors)
      - Note: Turbidity is binary

## Subfolder: model_assumptions_and_scenarios

- File: JSON file with assumptions related to the cashflow model
    - Name: cashflow_rate_assumptons.json
- File: JSON file with assumptions related to the infrastructure financing model, including infrastructure options (costs, deployment time, etc.)
    - Name: inf_planning_assumptions.json
- File: Csv file with water resources model assumptions involving single values
    - Columns: Assumption (name of assumption), SCWSM-Option_Analysis… (version name, most are based on the demand scenario, examples include: SCWSM-Demand_Baseline)
- File: Excel file with water resources model assumptions requiring monthly profiles
    - Columns: Assumption (name); 1, 2, … (month of year)
    - Tabs: Names for each version, i.e., SCWSM-Option_Analysis
- Files created in simulation process:
    - used_assumptions_PROFILES.csv: contains the monthly profiles utilized in the model
    - used_assumptions_CSTE.csv: contains the single values utilized in the model
- File: JSON file with sensitivity analysis-relevant parameters
    - Name (example): file_SA.json
    - Notes: The code has been designed to be able to input an empty name for this file or a file with any combination of the possible sensitivity analysis parameters. In this example, all filenames are the baseline files, except resampled_households.csv, which is not included for data privacy reasons.
- File: Csv file containing the values for the hydrological condition type by month of year
    - Columns: Month, Driest (5), Dry (4), Normal (3), Wet (2), Wettest (1)

## Subfolder: model > SCWSM-Option_Analysis > Parent

- File: JSON file containing the parameters used to build the Pywr model
    - Name: CST_SCWSM_OPTION_ANALYSIS_SA.json
    - Notes: Parts of this file have been removed including all nodes, all edges, some parameters, and some recorders
- Note: create subfolder to save individual json files into for stochastic simulations

## Subfolder: results

- File: Csv vile continuing the Loch Lomond quantile levels based on the month of the year
    - Columns: Date; Level (MG) 0.1, 0.15, 0.2, 1.0

## Subfolder: scripts

- File: Python file that sets up the Pywr model for that specific simulation, including importing the climate data and the sensitivity analysis parameters
  - Name: Setup_SCWSM_Option_Analysis_CST.py
- File: Python file that defines the class householdDemand
  - Name: household_demands.py
- File: Python file that contains all of the parameter classes input into Pywr built for this project and paper
  - Name: scwsm_demands.py
- File: Python file with the parameter classes for the alternative water supply options and operations
  - Name: scwsm_alternative_supply_option_operations.py
  - Parameter classes for specific water resources system removed, including:
    - SUPPLY_GAP: specific to SCWD
    - FELTON_DIV_TO_LL: specific to SCWD
    - DESALINATION_PLANT_YIELD/DESALINATION_PLANT_YIELD_ROF,: determines the desalination plant yield based on the month of the year and storage levels
    - DIRECT_POTABLE_REUSE_PLANT_YIELD: determines the DPR plant yield based on the month of the year and storage levels
    - DEMAND_AFTER_RATIONING: determines the value of reduced urban demand depending on if rationing/curtailment is implemented in the model
    - TRANSFER_YIELD: returns the transfer volume from x neighboring water district based on reservoir levels
    - MCGWB_SWITCH: a switch that turns the capacity of extraction to 0 on May 1st based on reservoir storage
    - Soquel_SWITCH: a switch that turns the supply from Soquel ASR on/off based on reservoir levels- checked between May and September
- File: Python file that imports assumptions from relevant csv files
  - Name: scwsm_model_assumptions.py
  - Note: Csv files imported not included in Github repository
- File: Python file that includes parameter classes for relevant operations and constraints
  - Parameter classes:
    - Max_flow_below_LL: sets the maximum flows downstream of the reservoir based on the month of the year
    - Hydraulic_constraint: calculates the hydraulic loss factor for a pump in the system
  - Note: not included in Github repository
- File: Python file that includes parameter classes for relevant water rights assumptions and operations
  - Parameter classes:
    - Felton_to_LL_Annual_License: calculates the annual license at a diversion within the system

        - WATER_RIGHT_NEWELL_TO_GHWTP: returns current value of reservoir license tied to use of upstream source for water supply
        - WATER_RIGHT_NEWELL_DIV_TO_LL: returns current value of license tied to certain diversion
        - PRECIP_WATER_IN_LL: returns the volume of water from local rainfall over the reservoir
        - FELTON_WATER_IN_LL: returns the volume of water in the reservoir originating from a certain source
        - NEWELL_WATER_IN_LL: calculates the volume of water in the reservoir from a specific upstream source
        - USEABLE_PRCP_TO_GHWTP: reuters volume available from precipitation in reservoir available for water treatment plant
        - WATER_MONTH_TYPE: returns the water month type based on flow from an uptream source
        - SWITCH_MCASR_INJECTION: returns the binary value based on the water month type parameter
        - SWITCH_TRANSFERS returns the binary value based on the water month type parameter
    - Note: not included in Github repository
- File: Python file that runs the simulations used in the paper
    - Name: Run_simulations_paper_Sherlock.py
    - Note: run on HPC system
- File: Python file that runs the infrastructure financing & rate design sensitivity analysis used in the SI
    - Name: Run_simulations_Sherlock_InfRates_SA.py
    - Note: run on HPC system
- File: Python file that runs the demand & demographics sensitivity analysis used in the SI
    - Name: Run_simulations_Sherlock_DemandDemographics_SA.py
    - Note: run on HPC system
- Subfolder: plots
    - File: Jupyter notebook file to create paper plot Figure 1
        - Name: Figure1_ModelFramework.ipynb
    - File: Python file to process the data for and create paper plot Figure 2
        - Name: Figure2_AddClimate_Sherlock.py
        - Note: this figure involves a lot of data and should be run on an HPC system
    - File: Python file to create paper plot Figure 3
        - Name: Figure3_AddClimate_Sherlock.py
        - Note: this figure involves a lot of data and should be run on an HPC system
    - File: Python file to process and plot Figure 4
        - Name: Figure4_Sherlock.py
    - File: Python file to process and plot Figure 5
        - Name: Figure5_Sherlock.py

- File: Python file with functions for processing the different types of data across results figure files
    - Name: processing_functions.py
- File: Jupyter notebook file to create boxplots for infrastructure financing and rate design sensitivity analysis (Figure S10)
    - Name: SensitivityAnalysis_InfRates.ipynb
- File: Jupyter notebook file to create boxplots for demands and demographics sensitivity analysis (Figure S10)
    - Name: SensitivityAnalysis_DemandDemographics.ipynb