Profile Plots Script: CE-QUAL-W2 Post-Processing and Visualization

Python Code developed by Cerrina Mouchref, Stephanie Meister, and Josh Gottlieb
Spring 2022

Overview

This script produces a profile plot for a water quality constituent for each common day in two datasets: an observed set of data for a water body and modeled output for the same location. The user inputs values into the configuration excel file to specify their parameters of interest, location of datasets (in .csv or excel format), plot attributes, and output destination folders. In addition to profile plots, the script creates an excel file of daily and whole-period error statistics including mean, mean error, absolute mean error, root mean squared error, standard deviation, and percent bias.

Reading the Code

The profile plot script can be used in Microsoft Visual Studio, Anaconda, or any other IDE compatible with Python software. Open "CEQUALW2_ProfilePlot" in your IDE. Several free and downloadable python modules are required to run the script. These are:

- Matplotlib
- Numpy
- Pandas
- datetime
- pathlib
- imageio
- openpyxl
- dateutil
- opencv-python

These modules can be downloaded using pip install in the command line of windows and mac computers. Detailed instructions on pip installing python modules can be found here: https://docs.python.org/3/installing/index.html

Folder Setup

The profile plots script requires a specific folder organization to both read in the datasets properly, and output the plots and .gif animation to the right place. The default folder structure for the script is shown below in figure F1:

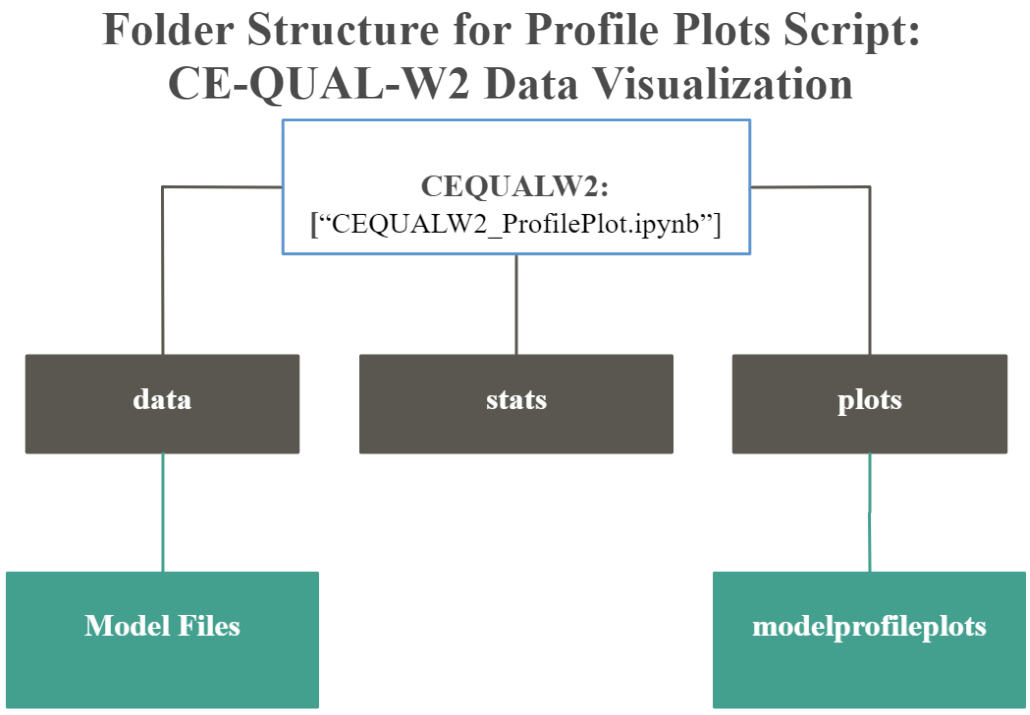

Figure F1: Default folder structure. The main folder "CEQUALW2" contains the python file for the script, as well as subfolders "data", "stats", and "plots".

These folders can be renamed by loading custom filepath addresses into configuration file, but the general structure should remain the same for simplicity. Keeping the python script in the CEQUALW2 folder and using this folder as the python directory in your IDE is encouraged.

User Input: Configuration File

User input for the profile plots code runs through the configuration file: *makeprofileplotconfig.xlsx*. Open the file and enter selections in columns 2 and 3 corresponding to the observed data and modeled data respectively.

$ Template config file for Profile plots		
$ Create a profile plot of 1 variable from observed and modeled datasets		
$ 4/30/22		
File	data\Model Files\HaggRes_InletStreams_WQ.csv	data\Model Files\spr_wb1.csv
Sheet Name		
Skip Rows		
Site Column Name	Site	
Site Name	V - Hagg Lake	N/A
Variable Column Name	Lab Parameter	Constituent
Variable Name	Temperature	Temperature
Units Column Name	Lab Units	N/A
Variable Units	deg C	N/A
Result Column Name	Result_as#2	Seg_29
Depth Column Name	Depth_m	Depth
NA Values	-999	-99
Date Column	JDAY	Julian_day
Julian Start Day	1/1/2013	
Legend Label	Observed	Modeled
Figure Title	Profile of Hagg Lake Segment 29 on	
X Label	Temperature, C	
X Axis (min, max)	0	25
Y Label	Depth, m	
Y Axis (min, max)	0	35
Profile Plots Folder	plots\modelprofileplots	
Statistic Output Folder	stats	

Figure F2: Configuration (Config) file.

Config file rows:

- File: Paste the filepaths for the observed and modeled datasets within the parent directory.
- Skip Rows: Specifies number of rows to skip in each dataset if there are blank headers. These cells can be left blank
- Site Column Name: Header name for column specifying particular site in observed data.
- Site Name: Specifies the specific site if the observed dataset includes multiple locations. Will usually be blank for the model dataset.
- Variable Column Name: Specifies the header name of the parameter column.
- Variable Name: Parameter that will be plotted, case sensitive.
- Units Column Name: Header for units column
- Variable Units: Units to be plotted on x axis. Case sensitive.
- Result Column Name: Column which displays the observed and measured values of the parameter
- Depth Column Name: Column listing depths.
- NA Values: The designation of null values for each dataset. Commonly -99 or -999.
- Date Column: Column listing date of entry.
- Julian Start Day: First day of Julian count, which will be converted to date.
- Legend Label: Label for each dataset to be included in plots.
- Figure Title: Title of plots. This entry will be followed by the date of each output.
- X Label: X axis label
- X Axis: Specifies the limits of the x axis.
- Y Label: y axis label
- Y Axis: Specifies the limits of the y axis.
- Profile Plots Folder: Filepath within the parent directory for storage location of output plots and animations.
- Statistics Output Folder: Filepath within parent directory for storage location of statistics .csv file.

Data Structure

The Profile Plots Script loads .csv or excel data files output from CE-QUAL-W2 using the filenames specified in the configuration sheet. Most CE-QUAL-W2 data is output in a “long” format, with each parameter value output in a new row, and multiple rows making up a single day of observations or model output. In contrast, “Wide” formatted data usually contains a single row for each day, with multiple columns for each parameter or measurement.

The current version of the Profile Plots Script is only compatible with long-formatted data. Wide data tables must be converted to long format using R or Python processing before the script can read the values, and an error value may interrupt the script.

Figure F3 shows the difference between wide and long formatted data.

Day	Depth	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
1	x	x	x	x	x	x
2	x	x	x	x	x	x
3	x	x	x	x	x	x

Day	Depth	Parameter	Value
1	x	x	x
1	x	x	x
1	x	x	x
1	x	x	x
1	x	x	x

Figure F3: (Above) Wide-formatted data table. Each parameter is specified in a particular column. Currently not compatible with the profile plots script. (Below) Long formatted data table. The current format compatible with the script, each day in the set is repeated in multiple rows, with each row storing a different parameter.

Column names are specified by the user in the configuration file, and should include the parameter of interest, the units, and column which stores the values of the parameter.

Executing the Code

With the inputs specified in the user config file, select “run all” in the IDE to execute all notebooks of the code. Check error outputs to ensure that calculations have eliminated null values in your dataset, and cross check the output plots to validate that data matches the visualization. Parameters can be adjusted between runs - make sure to change the plot names for the outputs to ensure that files are not overwritten in consecutive runs.