Statistical Analysis Documentation

Elise Jackson

# 4/2/18

# Metrics Documentation

Determining the metrics to use in validating models is a function of the individual agency and region. The most common metrics used in the validation of hydrologic models are:

* Correlation Coefficient
* Mean Difference
* RMSE
* RMSLE
* NSE
* R2 coefficient

For a full summary of these metrics, and how they react to different trends within hydrographs, see the following documents:

[Metrics Summary Powerpoint](https://drive.google.com/open?id=1EQPqjqghkBeAzykTBOf6SHhp5F4sTKOy): PowerPoint presentation summarizing the main metrics used, and the advantages of each.

The Hydrostats python package contains over 50 different metrics that can be used to analyze and validate two time-series.

The full documentation for the Hydrostats Python package can be found [here](https://waderoberts123.github.io/Hydrostats/).

However, this training focuses on using seven metrics that were determined to be the most effective in analyzing the differences and similarities between two time-series. Other metrics can be calculated using the functionality within the Hydrostats Python package.

# Getting Started

The Jupyter Notebooks discussed in this documentation contain several workflows that calculate specific metrics for multiple stations. These workflows can be customized for the individual agencies and regions, to reflect the stations, and specific metrics of interest for each agency.

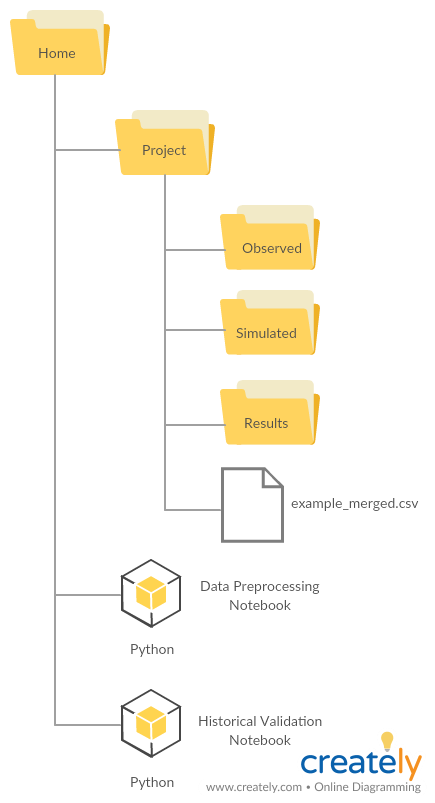
## Computer Setup

Before these workflows can be used on a computer, the required Python environment and tools must be installed:

1. Install the latest Python Environment (must be 3.5 or higher)
   1. Follow the instructions [here](http://hydroinf.groups.et.byu.net/tethys/conda/) to install:
      1. Miniconda
      2. Python 3.x (in the instructions you can skip over setting up Python 2.7)
2. Install Jupyter Notebooks
   1. Follow the instructions [here](http://hydroinf.groups.et.byu.net/tethys/jupyter/) to install and setup a directory on your computer to store your notebooks.
   2. Save the Data Preprocessing and Historical Validation notebooks into this directory.
3. Create a file structure

These workflows require a specific file structure so that the script knows where the different streamflow files are located, as well as knows where to save the validation results at the end of the calculations.

Create a folder where the data files from the time series will be saved. This is where the results from the validation will also be saved. Remember this directory, as it will be used as inputs in the notebook.



Create a folder named after the region that is being analyzed (‘Project’ in this example). The .csv files with the time series to be compared will be saved in that folder.

1. Launch Jupyter
   1. Use the instructions found [here](http://hydroinf.groups.et.byu.net/tethys/jupyter/#launching-jupyter) to launch Jupyter notebooks.
   2. To preprocess the streamflow data prior to validation, open the Data Preprocessing Notebook.
   3. If the data has already been formatted, open the Historical Validation Notebook

Jupyter is a free and open source application that runs in a web browser. You enter lines of Python code in blocks called "cells". Each cell can contain a single line of code, or several lines of code. You can execute the code in a cell by putting the cursor in the cell and hitting Shift-Enter. The output from that cell (if any) is then displayed directly below the cell. This allows you to incrementally develop your code and preserve the code and the output in a notebook file.

# Data Preprocessing Notebook

This notebook contains the code necessary to preprocess the stream flow data prior to validation. This notebook is available through the Google Drive folder. Download the notebook, and store it in the directory previously setup as part of the Jupyter installation process. This notebook consists of four main segments of code:

1. Install Modules
2. Download simulated data from SPT
3. Access Observed data from Hydroserver
4. Merge simulated/Observed Data

The last three code segments can be run independently, however the first segment “Install Modules” must be run prior to the others, as it installs the appropriate modules for the code to run.

## Identify Station

Accessing the simulated data can be done through the SPT portal, or through the API. We will discuss both methods in this documentation.

### SPT Portal

Identify the station at which you wish to conduct your statistical analysis. For an accurate comparison between the simulated and observed data, the observed data should consist of daily values for a time period of at least 35 years. Observed data should be in a .csv format, with the first column as date, and the second column as flow, as shown in Figure 1.

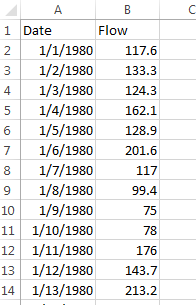
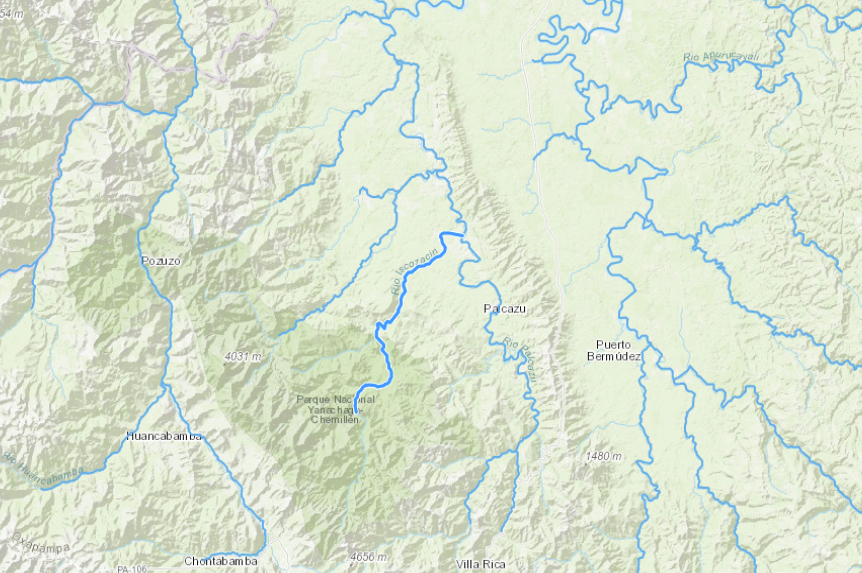


Figure : Example of observed data in a .csv file.

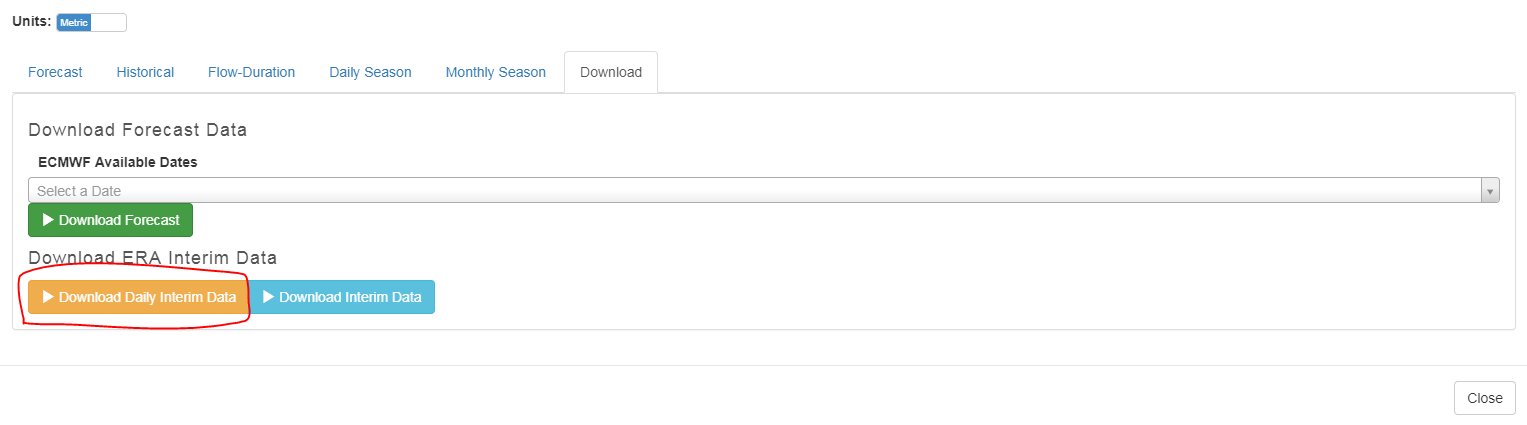
After the station has been identified, the SPT can be used to download the simulated data for the station. Using the SPT map window to identify the corresponding stream reach to the station, click on the stream reach. The selected stream reach will then be highlighted within the SPT map window.



Click on the “Historical” tab to view the 35-year simulated dataset for the station.



Click on the “Download” tab, then the “Download Daily Interim Data” tab to download the 35-year simulated dataset as a .csv file.

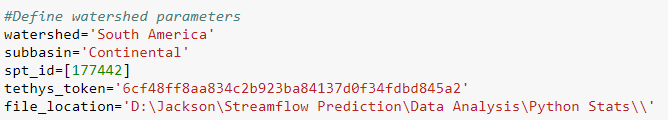


### SPT API Access

The SPT data can also be accessed through the SPT API. To use this API, the user must specify the following variables:

1. watershed = the name of the watershed within the SPT
2. subbasin = the name of the subbasin within the SPT
3. spt\_id = the ID number of the streamreach where the station is located. This can be a single station ID, or multiple ID’s in a list format.
4. tethys\_token = this token is available through the settings of the Tethys portal
5. file\_location = file location to save the simulated data as a .csv file

Input these variables inside the section of the Jupyter notebook corresponding to “Access SPT Data”.

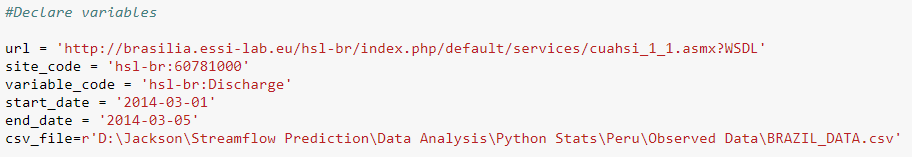


To run the workflow inside of the Jupyter notebook, click inside the code block, and press “ctrl+Enter”. This will run the workflow, and save the downloaded .csv files within the file location variable previously defined.

## Access Observed Data from HydroServer

Observed data can be accessed directly from agencies, or can be accessed through the API of a Hydroserver. In order to access data through the API, the user must specify the following variables:

1. url = the url endpoint for the Hydroserver where the observed data is stored
2. site\_code = the hydroserver-specific code that corresponds to the streamreach in question
3. variable\_code = the data-specific code that corresponds to the data (such as discharge) that the user wants to download.
4. start\_date = the beginning of the time to be downloaded. Must be in the format 'YYYY-MM-DD'
5. end\_date = the beginning of the timeframe to be downloaded. Must in the format 'YYYY-MM-DD'
6. csv\_file = the file location where the downloaded data will be saved as a .csv file



This workflow then accesses the data saved on the Hydroserver, and downloads it as a csv file to the specified directory.

## Merge Simulated/Observed Data

The simulated data must then be merged with the observed data to be used. The next section in the Jupyter notebook merges these datasets.

To use this workflow, the user must specify:

1. recorded\_dir = directory where the .csv files with the observed flow data are saved
2. interim\_dir = directory where the .csv files from the SPT are saved
3. merged\_dir = directory where the merged .csv files will be saved. This directory should be the up one level from where the observed and SPT data are saved.

# Historical Validation Notebook

The Historical Validation notebook is a Jupyter Notebook that contains the Python code necessary to run the correlation and lag analysis workflows. This notebook is available through the Google Drive folder. Download the notebook, and store it in the directory previously setup as part of the Jupyter installation process.

Inside of the notebook, specific blocks of code can be run inside of cells using “ctrl+Enter”. The resulting .csv files will be stored in locations specified through variables by the user.

This notebook contains two major workflows:

1. Correlation Analysis Workflow
2. Lag Analysis Workflow

The Correlation Analysis workflow has been divided into separate functions, to help users better understand the order of operations within the workflow. All of these code blocks must be run in order to successfully validate the data.

# Workflow Documentation

The Correlation Analysis workflow, as well as the Timing Metrics workflow were written to calculate the seven standard metrics used most often in hydrology, as well as calculate the R2 coefficient and spectral angle for the two time series. These workflows can be changed to reflect the needs and special concerns for individual regions, or can serve as a base for personal workflows for the different regions.

To successfully run the workflows, the machine must be configured based on the guidelines in the [Computer Setup](#_Computer_Setup) section of this document.

These workflows are also designed to work with a merged .csv file. If the user has used the Data Preprocessing Notebook, the merged files have already been formatted appropriately to best work with the Historical Validation Notebook.

If the user did not use the Data Preprocessing notebook to format the data, certain conventions must be followed, such as:

* File name: must end with ‘\_merged.csv’. This allows the functions to recognize the csv file as a merged data file.
* Data format: the merged file must be created with the first column of Date, the second column of predicted streamflow, and a third column of observed streamflow.

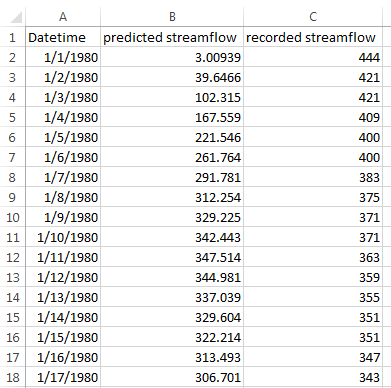
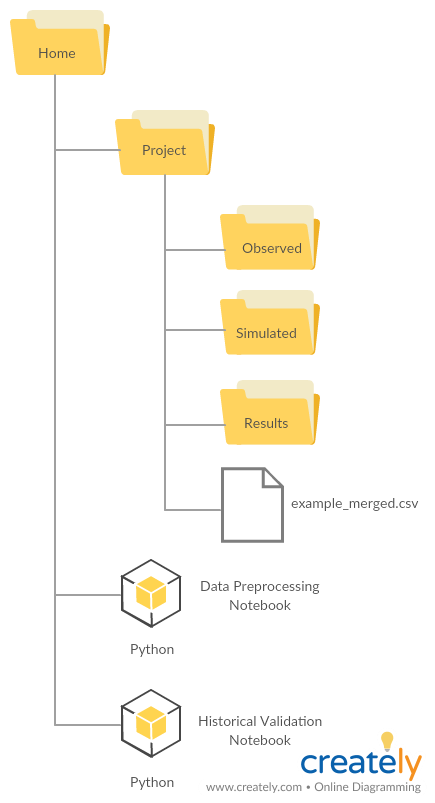


Figure : Example of merged dataset formatting.

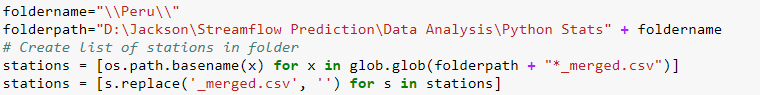
## Correlation Analysis Workflow

This workflow calculates the standard metrics on a monthly and yearly basis by averaging the 35-year history in monthly and yearly increments. The workflow then produces a folder titled “Results”, which contains both the individual csv’s for the stations, as well as a national summary .csv file with all the data combined. This file can then be used to create tables and compare the different stations. A folder titled “Plots” is also created. This folder contains the average yearly hydrographs for each of the individual stations, comparing the simulated and observed data values.

To run the script, the following data structure must be created in your directory. If you previously created these folders, you don’t have to do it again. In the project folder, save the Correlation Analysis.py, as well as a new folder with your merged data files. Give this folder a name (in this example, “Project”) that will later be used as a variable in the script.

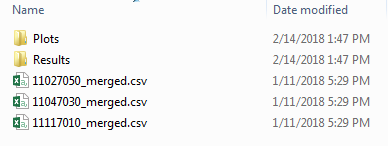


The first several blocks of code are the individual functions inside of the workflow. These explain the process that the workflow uses to validate the data. To validate your data, set the variable “foldername” to the name of the folder previously created. The variable “folderpath” must also be updated to correspond to where the data were saved.

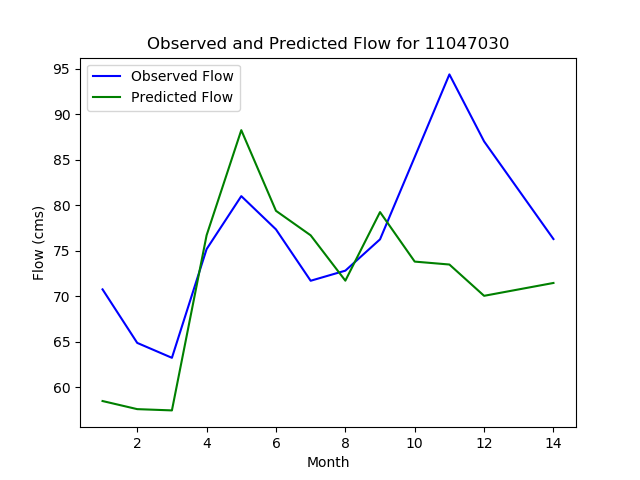


This tells the script where to look for the merged data files. The script will then cycle through each merged .csv file and calculate the metrics for each.

Inside the data folder, two new subfolders have been created “Results” and “Plots”



The Plots folder contains a hydrograph for each station, comparing the simulated and observed data values.



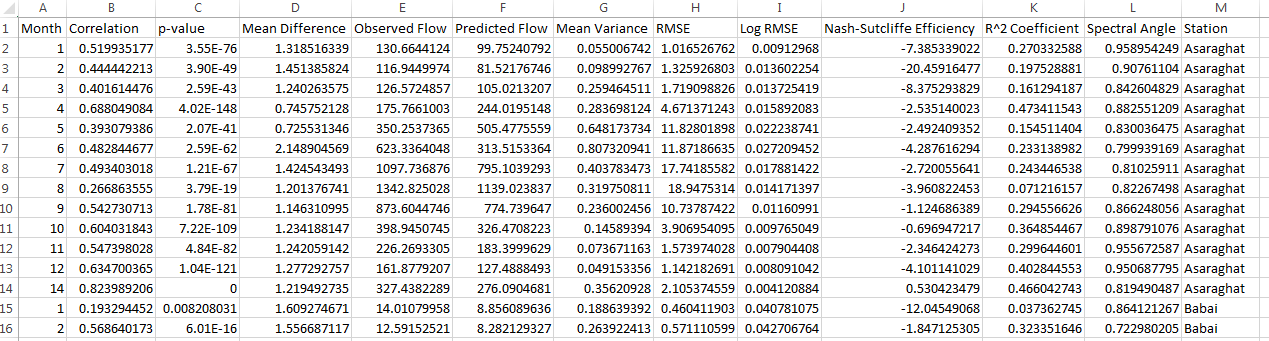
The Results folder contains a .csv for each individual station, as well as a “National Results.csv” file, which is the summary of each station run.



This results file contains the summary for the following metrics:

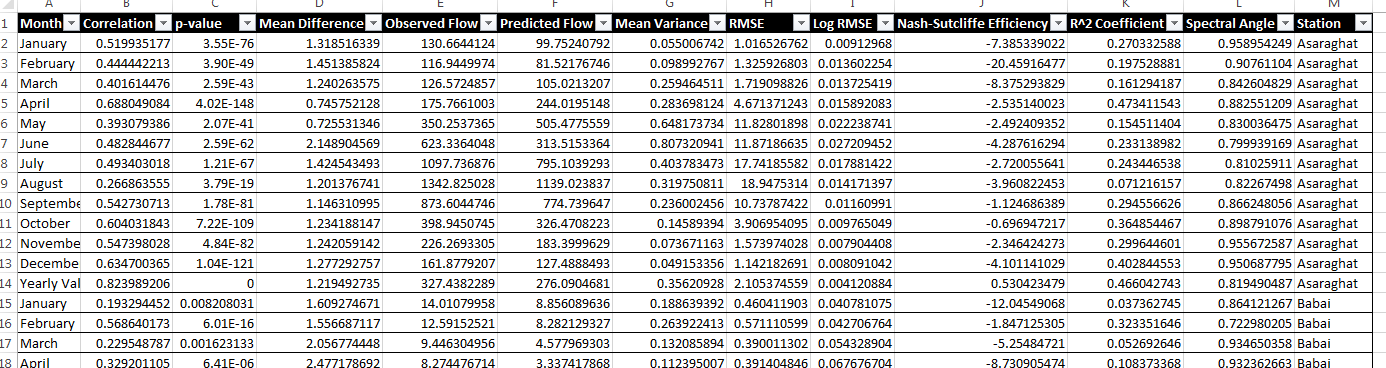
* Correlation Coefficient
* Mean Difference
* Observed Flow
* Predicted Flow
* Mean Variance
* RMSE
* RMSLE
* NSE
* R2 coefficient
* Spectral Angle coefficient

Opening the National Results.csv file allows the user to see all the statistics, on a monthly and yearly basis for each station.



The numbers 1-12 in the Month column correspond to the monthly values, while the 14 corresponds to the Yearly metric value.

To format the data to be easier to access, highlight all the data and click “Format as Table”. Choose one of the predetermined formatting choices, and make sure that the “My table has headers” box is checked. Then in the Month column, replace the numbers with the names of the month and “Yearly Value” for the 14. After the formatting has been done, your table should look similar to the one below.



This data table can then be used to create graphs and tables summarizing the error trends across the different months and stations.

## Lag Analysis Workflow

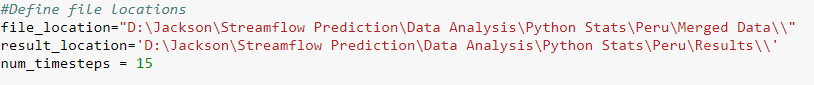
To calculate the timing tests on the data, the Lag Analysis workflow should be run from the Jupyter notebook. This workflow analyzes the need for lagging the data to improve the overall correlation for the R^2 coefficient and the spectral angle coefficient.

To use this workflow, the user must specify where the merged data files are saved, as well as where the .csv results file from the lag analysis will be saved.

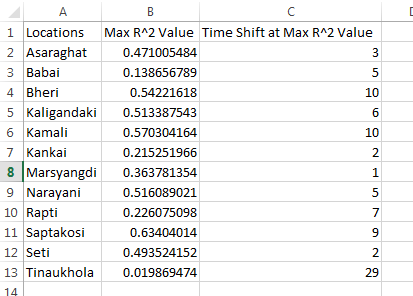
Create a new folder in the project folder called “Merged Data”, where a copy of each .csv file of the merged flows are saved.

The user must input:

1. file\_location = directory where the merged data files are saved. This should be a specific folder of files.
2. result\_location = directory where the results from the analysis will be saved.
3. num\_timesteps = number of timesteps for the lag analysis to consider. A timestep is equal to 6 hours.



Run the cell, and two new files will be created in the “Results” folder, “r\_squared\_summary\_NP.csv”, and “Spectral\_Summary\_NP.csv”.



This file summarizes the maximum R2 coefficient value, as well as the number of time steps required to produce that value. Each time step is equal to a time lag of 6 hours. The Spectral\_Summary file shows the same summary, but for the spectral angle coefficient.

## Data Summary

The scripts used in these statistical analysis create basic .csv files with the different error metrics calculated. For presentation, these .csv files can be used to create simple tables and charts describing the error metrics and trends for each region.