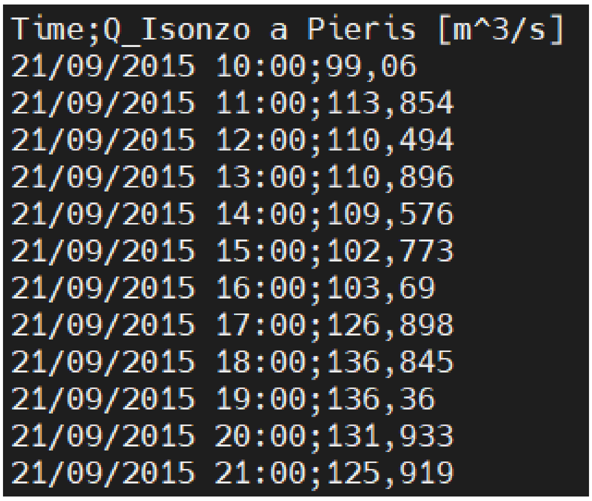
HOW TO USE THE SCRIPTS FOR PROCESSING RIVERS' DISCHARGES DATA

April 22, 2021

This document is aimed at explaining how to properly use the scripts developed to process (convert, extend, merge, correct and visualize) the time series of rivers’ discharges, provided by Civil Protection of Friuli-Venezia Giulia region (henceforth Civil Protection of FVG) and CNR-ISMAR.
The scripts briefly described below should be executed in the order in which these are listed.

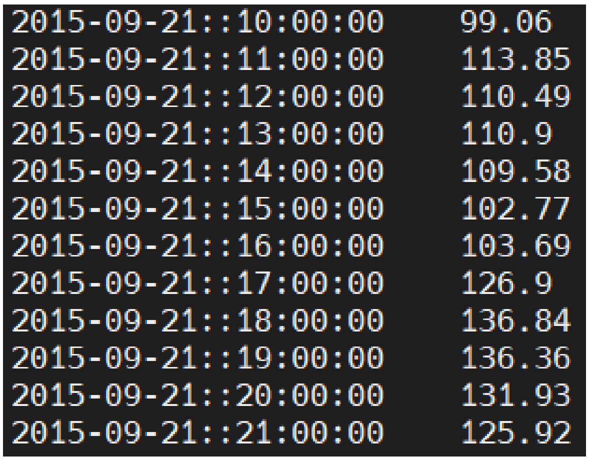
1) pciv_2_shyfem-rivers.py

This *Python* script is aimed at converting the format of the files containing the time series of rivers' discharges, as provided by Civil Protection of FVG (e.g. see Figure 1), to the format required in input by the SHYFEM model (e.g. Figure 2).
It is assumed that the time series to be considered are characterized by a constant time resolution, to be set in the script (see figure 3).
Possible time gaps (namely consecutive time steps separated by integer multiples of the time resolution) are filled by the introduction of the missing dates, associated with rivers’ discharges indicated with *nan* (NaN, Not a Number). For example, if an hourly-resolution time series is characterized by two consecutive time steps, separated by three hours (time gap), the two missing hours (required to have an actual constant resolution) are introduced, associated with rivers’ discharges indicated with *nan*.



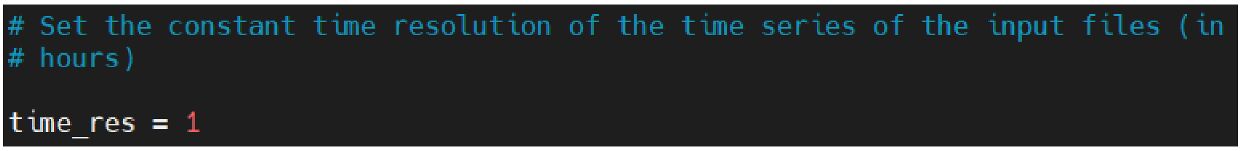
```
Time;Q_Isonzo a Pieris [m^3/s]
21/09/2015 10:00;99,06
21/09/2015 11:00;113,854
21/09/2015 12:00;110,494
21/09/2015 13:00;110,896
21/09/2015 14:00;109,576
21/09/2015 15:00;102,773
21/09/2015 16:00;103,69
21/09/2015 17:00;126,898
21/09/2015 18:00;136,845
21/09/2015 19:00;136,36
21/09/2015 20:00;131,933
21/09/2015 21:00;125,919
```

Figure 1 Example of time series of river’s discharges, as provided by Civil Protection of FVG.



```
2015-09-21::10:00:00    99.06
2015-09-21::11:00:00    113.85
2015-09-21::12:00:00    110.49
2015-09-21::13:00:00    110.9
2015-09-21::14:00:00    109.58
2015-09-21::15:00:00    102.77
2015-09-21::16:00:00    103.69
2015-09-21::17:00:00    126.9
2015-09-21::18:00:00    136.84
2015-09-21::19:00:00    136.36
2015-09-21::20:00:00    131.93
2015-09-21::21:00:00    125.92
```

Figure 2 Example of time series of river’s discharges, as required in input by the SHYFEM model.



```
# Set the constant time resolution of the time series of the input files (in
# hours)

time_res = 1
```

Figure 3 Where and how to set the constant time resolution of the time series to be considered.

2) *pciv_time_series_extension-rivers.py*

This *Python* script is aimed at extending a time series of river's discharges to a desired time period fictitiously, by the introduction of the missing dates, associated with river's discharges indicated with *nan* (NaN, Not a Number). For example, if it is desired to extend a time series, ranging from 2nd to 30th January, to the entire month of January, the dates 1st and 31st January will be properly introduced, associated with river's discharges indicated with *nan*.

It is assumed that the time series to be considered are characterized by a constant time resolution, to be set in the script (like for script 1); e.g. see Figure 3), and no time gaps. Moreover, it is assumed that the input files are formatted as required in input by the SHYFEM model (e.g. see Figure 2). In order to meet these requirements, script 1) should be used.

3) *pciv_merge-rivers.py*

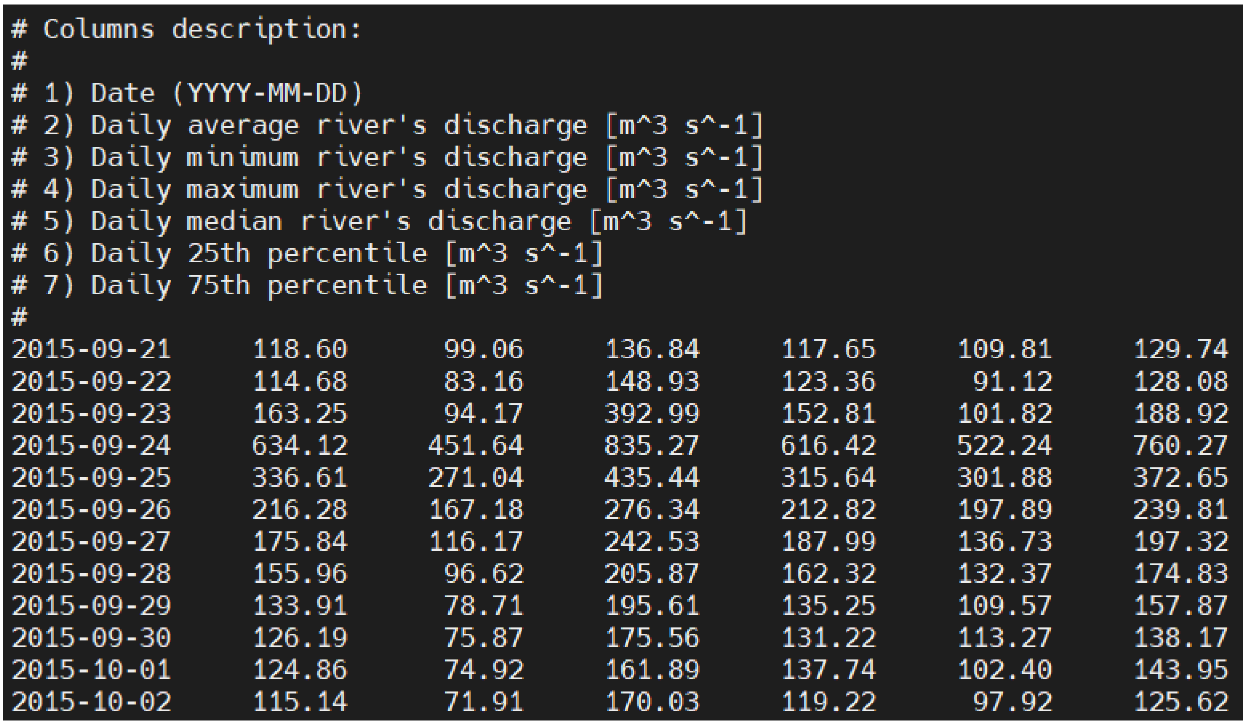
This *Python* script is aimed at merging two time series (of discharges of the same river), read from two different input files, properly. This is carried out according to the simple algorithm described below:

- if for a certain time step, only the data related to the first time series read are available (not NaNs), these are chosen;
- otherwise, the data related to the second time series are chosen (available or NaNs).

It is assumed that the time series to be merged are characterized by a constant time resolution, no time gaps and the same length. Moreover, it is assumed that the input files are formatted as required in input by the SHYFEM model. In order to meet these requirements, script 2) should be used.

4) *daily_statistics-rivers.py*

This *Python* script is aimed at computing some of the main statistical parameters (average, minimum, maximum, median and 25th and 75th percentiles) for a daily distribution of data (time series of river's discharges), characterized by sub-daily (or daily, as a boundary case) time resolution (e.g. see Figure 4). This is performed for the entire length of the time series to be considered.
It is assumed that the input files are formatted as required in input by the SHYFEM model. In order to meet these requirements, scripts 1), 2) or 3) should be used.



```
# Columns description:
#
# 1) Date (YYYY-MM-DD)
# 2) Daily average river's discharge [m^3 s^-1]
# 3) Daily minimum river's discharge [m^3 s^-1]
# 4) Daily maximum river's discharge [m^3 s^-1]
# 5) Daily median river's discharge [m^3 s^-1]
# 6) Daily 25th percentile [m^3 s^-1]
# 7) Daily 75th percentile [m^3 s^-1]
#
2015-09-21     118.60      99.06     136.84     117.65     109.81     129.74
2015-09-22     114.68      83.16     148.93     123.36      91.12     128.08
2015-09-23     163.25      94.17     392.99     152.81     101.82     188.92
2015-09-24     634.12     451.64     835.27     616.42     522.24     760.27
2015-09-25     336.61     271.04     435.44     315.64     301.88     372.65
2015-09-26     216.28     167.18     276.34     212.82     197.89     239.81
2015-09-27     175.84     116.17     242.53     187.99     136.73     197.32
2015-09-28     155.96      96.62     205.87     162.32     132.37     174.83
2015-09-29     133.91      78.71     195.61     135.25     109.57     157.87
2015-09-30     126.19      75.87     175.56     131.22     113.27     138.17
2015-10-01     124.86      74.92     161.89     137.74     102.40     143.95
2015-10-02     115.14      71.91     170.03     119.22      97.92     125.62
```

Figure 4 Example of output file produced by the script 4).

5) *pciv_bias_correction-rivers.py*

This *Python* script is aimed at computing and plotting, for a given range of degrees, the best-fit polynomial function *f* (least squares polynomial fit), between two samples x_d and y_d (time series of daily average river's discharges, provided by two different data sources and read from two different input files), both on a linear and semi-logarithmic scale (e.g. see Figure 5). In order to help the user in finding the best-interpolating function, among those considered, a Kolmogorov-Smirnov statistical test is performed (e.g. see Figure 6).

Moreover, histograms are built for the investigated distributions, either before and after the application of *f* to the sample x_d (e.g. see Figure 7).

Finally, assuming that *f* can be applied also to hourly distributions, although it is obtained from daily averages, the hourly time series x_h (read from a third input file) related to x_d, is corrected through this function (this script was born to bias-correct Civil Protection of FVG rivers' discharges, which are accurate during floods, but overestimated otherwise).

The input files containing the samples x_d and y_d should be produced through the script 4).

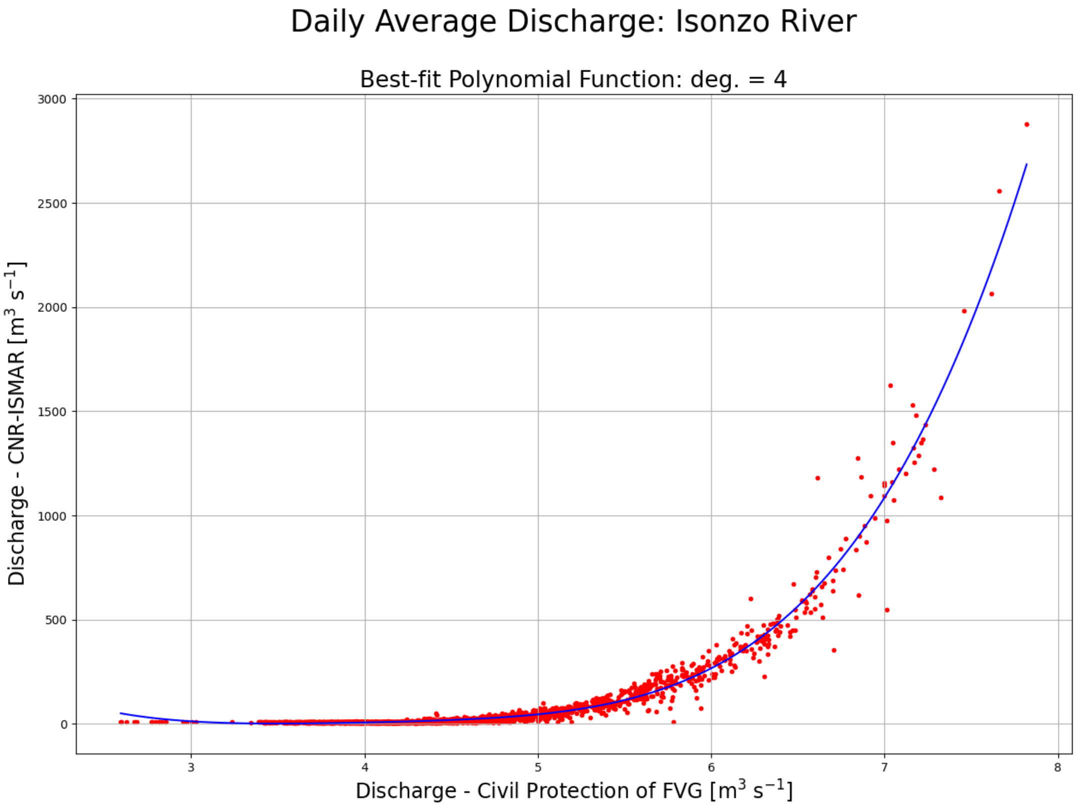
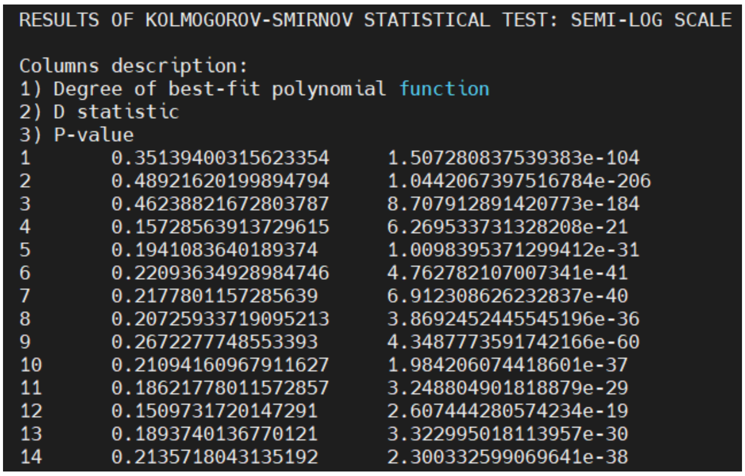


Figure 5 Example of best-fit polynomial function (least squares polynomial fit), on a semi-logarithmic scale, interpolating daily average discharges of the Isonzo river, as provided by Civil Protection of FVG (x-axis) and CNR-ISMAR (y-axis), for the time period ranging from September 21, 2015 to February 27, 2021.



```
RESULTS OF KOLMOGOROV-SMIRNOV STATISTICAL TEST: SEMI-LOG SCALE

Columns description:
1) Degree of best-fit polynomial function
2) D statistic
3) P-value
1        0.35139400315623354     1.507280837539383e-104
2        0.48921620199894794     1.0442067397516784e-206
3        0.46238821672803787     8.707912891420773e-184
4        0.15728563913729615     6.269533731328208e-21
5        0.1941083640189374      1.0098395371299412e-31
6        0.22093634928984746     4.762782107007341e-41
7        0.2177801157285639      6.912308626232837e-40
8        0.20725933719095213     3.8692452445545196e-36
9        0.2672277748553393      4.3487773591742166e-60
10       0.21094160967911627     1.984206074418601e-37
11       0.18621778011572857     3.248804901818879e-29
12       0.1509731720147291      2.607444280574234e-19
13       0.1893740136770121      3.322995018113957e-30
14       0.2135718043135192      2.300332599069641e-38
```

Figure 6 Example of standard output for the Kolmogorov-Smirnov statistical test, produced by the script 5).

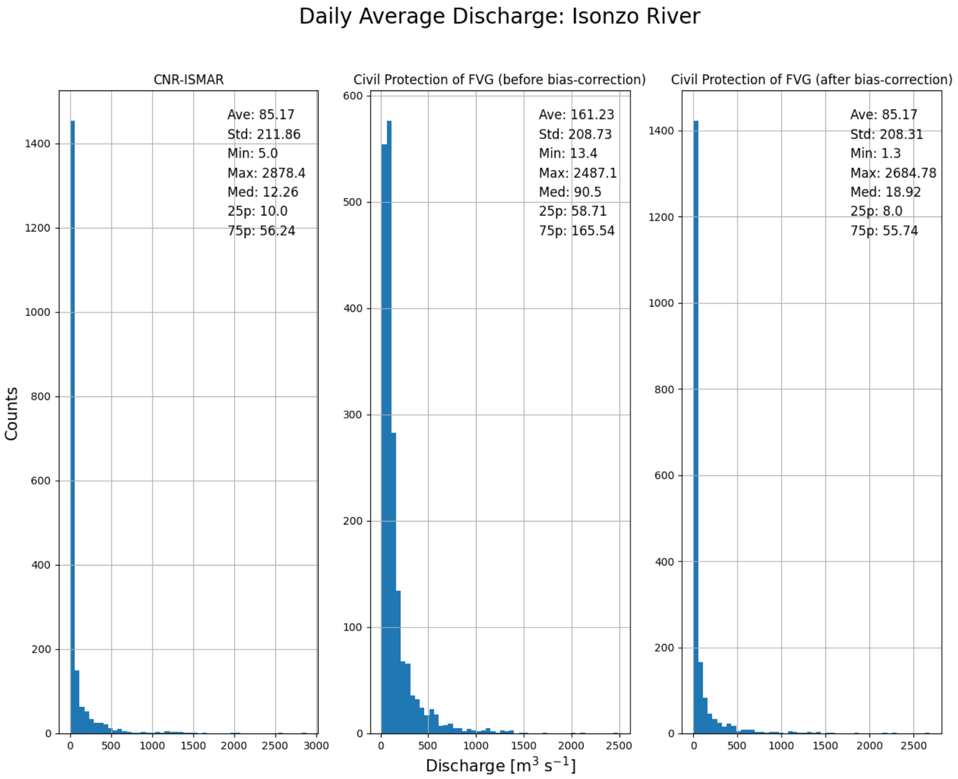


Figure 7 Example of plot produced by the script 5): histograms depicting the CNR-ISMAR (left) and Civil Protection of Friuli-Venezia Giulia region (before bias-correction, middle; after bias-correction, right) distributions of daily average discharges of the Isonzo river, for the time period ranging from September 21, 2015 to February 27, 2021. The values of some of the main statistical parameters (average, standard deviation, minimum, maximum, median, and 25th and 75th percentiles), for the entire time period investigated, are reported on the graphs.

6) *ismarcnr-pciv_boxplots-rivers.py*

This *Python* script is aimed at drawing box and whiskers plots (henceforth boxplots) and averages from pre-calculated daily statistics of distributions of river's discharges: a figure is produced for each month to be considered, and a boxplot is drawn for each day belonging to this month, together with the related daily average value. Moreover, also the corresponding average values, belonging to a different, comparison distribution, are plotted on the same figure (e.g. see Figure 8).
The input files containing pre-calculated statistics should be produced through the script 4).

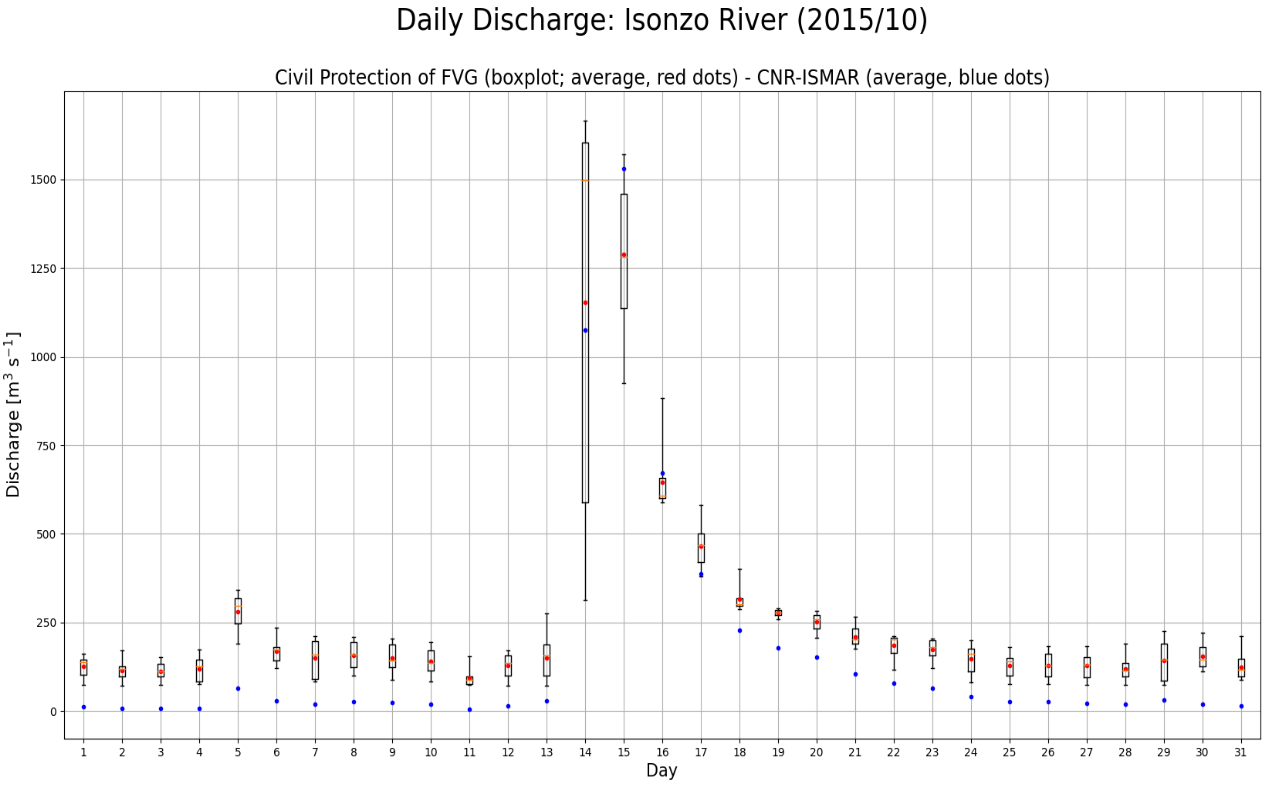


Figure 8 Example of plot produced by the script 6): daily discharges of the Isonzo river during the month of October 2015, as provided by Civil Protection of Friuli-Venezia Giulia region (daily box and whiskers plots and averages, red dots) and CNR-ISMAR (daily averages, blue dots). Boxes extend from the 25th to the 75th percentile, while whiskers extend till the minimum and maximum daily river's discharges; the median is depicted within the boxes through a horizontal, orange line.