PythOm: A Python toolbox implementing recent advances in rainfall intensity (Ombrian) curves

Theano Iliopoulou and Demetris Koutsoyiannis

National Technical University of Athens
Department of Water Resources and Environmental Engineering

- PythOm toolbox soon to be uploaded at: https://www.itia.ntua.gr/en/docinfo/2111/

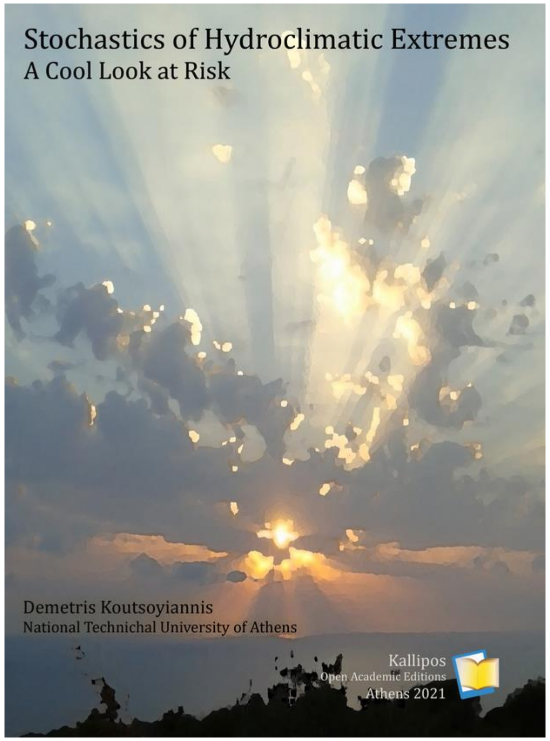
Motivation: recent advances in ombrian curves

Curves of rainfall intensity at various scales and for various return periods, else known as ombrian (or IDF) curves, are central design tools in hydrology and engineering.

- Construction of such curves often relies heavily on empirical or semi-empirical approaches, which hinder their applicability over large scales, and preclude simulation.

Recent work (Koutsoyiannis, 2021) has advanced these curves to theoretically-consistent stochastic models of rainfall intensity (**ombrian models**) extending their applicability to the full range of available scales, e.g. from minutes to decades.

See: Chapter 8: Rainfall extremes and ombrian modelling in *Stochastics of Hydroclimatic Extremes - A Cool Look at Risk,* ISBN: 978-618-85370-0-2, 333 pages, Kallipos, Athens, 2021. http://www.itia.ntua.gr/en/docinfo/2000/



Implementing the framework in Python environment

- We develop an open-source python toolbox (PythOm) implementing these advances in a straightforward and user-friendly manner.

- The toolbox also employs advanced estimation procedures from Koutsoyiannis (2021) including:
  - sophisticated statistical fitting methods for extremes (K-moments),
  - handling of bias induced by temporal dependence, and
  - optional blending of daily-scale data to reduce uncertainty of sub-daily records.

✓ The end result is the parameterization of the ombrian model and the graphical representation of rainfall intensity for any range of scales (supported by the data) and return periods.

From ombrian curves to ombrian models

Typical ombrian curves are advanced to stochastic models of the all-scale rainfall intensity, i.e. *ombrian models*. The ombrian model offers:

- mathematical and physical consistency and coverage of all time scales, from zero to infinity;
- provision for estimation bias due to time dependence;
- good behaviour on both very fine time scales and very large time scales, whereas conventional curves have a limited range of applicability;
- simultaneous treatment and preservation of the process's second- and higher-order properties along with the probability dry/wet.
- capability to perform direct simulation of rainfall intensity.

❖ These advances can be achieved on the basis of simple stochastic characterizations of the parent process, namely of its joint second-order and marginal higher-order properties.

Overview of modelling framework (I)

A theoretically-consistent ombrian model should ideally satisfy the following requirements (Koutsoyiannis, 2021):

- As in every stochastic model, the first and second order properties of the process of interest, i.e. the temporal average of rainfall intensity $\underline{x}^{(k)}$over any time scale k, should be preserved.

- The process's asymptotic variance at $k \to 0$ should be finite; the contrary would imply that the process requires infinite energy to materialize which is absurd for physical processes. In addition, the process's asymptotic variance at $k \to \infty$ should be zero, in order for the process to be ergodic.

- The model should deal with the intermittence of rainfall occurrences at fine time scales, describing both the probability dry $P_0^{(k)} \coloneqq P\{\underline{x}^{(k)} = 0\}$, and the probability wet, $P_1^{(k)} \coloneqq \overline{F}^{(k)}(0) = 1 - P_0^{(k)}$ for any time scale k, including for $k \to 0$.

- The principle modelling focus is on rainfall maxima, and hence it is important to preserve the higher-order properties of the process.

- The tail index of the rainfall intensity distribution should be constant for all time scales.

Overview of modelling framework (II)

- **At small time scales** the rainfall intensity follows a mixed type distribution, with a discrete part at the origin described by the probability dry, and a continuous part following the Pareto distribution with a constant tail index ξ and a state scale parameter $\lambda(k)$ as a function of the timescale:

$$F^{(k)}(x) = 1 - P_1^{(k)} \left(1 + \xi \frac{x}{\lambda(k)}\right)^{-1/\xi}$$

  - ❖ The Pareto distribution constitutes an optimal choice for small time scales due to its simplicity and explicit relationship between the time-averaged intensity and return period, and support by worldwide empirical evidence.

- **At larger time-scales** the rainfall intensity follows the Pareto-Burr-Feller (PBF) distribution with discontinuity at zero, characterized by an extra parameter $\zeta(k)$ as a function of the timescale:

$$F^{(k)}(x) = 1 - P_1^{(k)} \left(1 + \xi \left(\frac{x}{\lambda(k)}\right)^{\zeta(k)}\right)^{-1/\xi}$$

  - ❖ The PBF distribution is chosen for large scales because, contrary to the Pareto, it becomes bell-shaped for increasing $\zeta(k)$ which is consistent to the behaviour of the rainfall intensity at large time scales (cf. the central limit theorem).

Ombrian model formulation

		All-scale ombrian model	
		Small scales (Pareto) $k \le k^* \ll k^*_{\max}$	Large scales (PBF) $k \ge k^*$
	$x\ for\ \xi > 0$	$\lambda(k)\dfrac{(P_1^{(k)}T\ /\ k)^{\xi}-1}{\xi}$	$\lambda(k)\left(\dfrac{(P_1^{(k)}T\ /\ k)^{\xi}-1}{\xi}\right)^{1/\zeta(k)}$
	$x\ for\ \xi = 0$	$\lambda(k)\ \ln(P_1^{(k)}T\ /\ k)$	$\lambda(k)\left(\ln\left(P_1^{(k)}T/k\right)\right)^{1/\zeta(k)}$
		Properties	
mean	$\mathrm{E}[\underline{x}^{(k)}]$	μ	
climacogram	$\gamma(k)$	$\lambda_1(1+(k/\alpha)^{2M})^{\frac{H-1}{M}}$ or $\lambda_1(1+k/\alpha)^{2H-2}+\lambda_2(1-(1+a/k)^{2H-2})$	
Probability wet across scale	$P_1^{(k)}$	$\dfrac{1-\xi}{1/2-\xi}\dfrac{\mu^2}{\gamma(k)+\mu^2}$	$1-(1-P_1^{(k^*)})^{\left(\frac{k}{k^*}\right)^{\theta}}$
Inverse of lower tail index function	$\dfrac{1}{\zeta(k)}$	1	$\sqrt{(1-2\xi)\left(P_1^{(k)}\dfrac{\gamma(k)+\mu^2}{\mu^2}-1\right)}$
Inverse of State-scale function	$\dfrac{1}{\lambda(k)}$	$\dfrac{P_1^{(\mathrm{k})}}{\mu(1-\xi)}$	$\dfrac{P_1^{(k)}}{\mu}\left(1+\dfrac{1}{(1-\xi)(\zeta(k))^2}-\dfrac{1}{(\zeta(k))^{\sqrt{2}}}\right)$

Step I: Identification of the second-order dependence structure

- **Using the climacogram stochastic tool:**

$$\gamma(k) \coloneqq \operatorname{var}\left[\frac{\underline{X}(k)}{k}\right]$$ where $\underline{X}(k)$ is the process $\underline{x}(t)$ aggregated at timescale k.

with two alternative 4-parameter models for the climacogram structure:

- Filtered HK Cauchy (FHK-C) type: $$\gamma(k) = \lambda_1 \left(1 + (k/\alpha)^{2M}\right)^{\frac{H-1}{M}}$$
- Filtered HK Cauchy-Dagum (FHK-CD) type for a rough and persistent process, and for the special case $M = 1 - H$: $$\gamma(k) = \lambda_1 \left(1 + \frac{k}{\alpha}\right)^{2H-2} + \lambda_2 \left(1 - \left(1 + \frac{\alpha}{k}\right)^{2H-2}\right)$$

where α and λ_1, λ_2 are scale parameters, with dimensions $[t]$ and$[x^2]$, H is the so-called Hurst parameter ranging in the interval (0,1) and M is a dimensionless parameter which controls the local scaling of the process (fractal behaviour).

Example Python implementation:

```
## climacogram models
# Filtered HK Cauchy type
def FHK_C(k,l1,a,M,H):
    g_k=l1*((1+(k/a)**(2*M))**((H-1)/M))
    return g_k

## Filtered HK Cauchy-Dagum-type (FHK-CD) for M=1-H (rough process)
def FHK_CD(k,l1,l2,a,H):
    g_k=l1*(1+k/a)**(2*H-2)+l2*(1-(1+a/k)**(2*H-2))
    return g_k
```

```
# Obtain a first estimate of climacogram parameters from climacogram fitting
# and choose climacogram model
x0_CD=[0.05, 2, 2, 0.7]
x0_C=[0.05, 2, 0.5, 0.7]

res_CD=minimize(CgCD_Error, x0_CD, args=(s_len,d_len,var_k,k_h,scales_h,scale
res_C=minimize(CgC_Error, x0_C, args=(s_len,d_len,var_k,k_h,scales_h,scales_d

if res_CD.fun<res_C.fun:
    print "best fitting CD"
    print "error:", str(res_CD.fun)
    print "parameters:", str(res_CD.x)
elif res_CD.fun>res_C.fun:
    print "best fitting C"
    print "error:", str(res_C.fun)
    print "parameters:", str(res_C.x)
```

- More details in Section 3.13 (SoE)

Step II: Identification of the probability wet/dry structure

- **Using a maximum entropy structure for the probability dry** (Koutsoyiannis, 2006):

$$\ln P_0^{(k)} = \ln P_0^{(k^*)} \, (k/k^*)^{\theta}, \qquad k \geq k^*$$

where k^* is the transition time scale from Pareto to PBF distribution, for which

$P_0^{(k^*)} > 0$ and $\zeta(k^*) = 1$, and θ is a parameter ($0 \leq \theta \leq 1$).

- The transition time scale k^* is chosen at a point where the deviation of probability dry derived from the Pareto model from the empirical one is marginally acceptable.

- Default value of 24 h.

➢ **Example Python implementation:**

```
def pw_error(params,s_len,d_len,k_h,scales_h,scales_d,Cg_type,pw_k):
    m=params[0]
    xi=params[6]
    l1=params[2]
    scales=scales_h+scales_d
    if Cg_type=='CD':
        l2=params[3]
        a=params[4]
        H=params[5]
    ## theoretical climacogram in hours
        th_var_k=pd.Series((FHK_CD(k_h,l1,l2,a,H)),index=scales)

...

    ## theoretical probability wet
    th_pw_k_P=((1-xi)/(0.5-xi))*((m**2)/(th_var_k+m**2)) #
    ## divergence:
    E_p=(th_pw_k_P-pw_k)**2
    E_totp=E_p.sum()
    return E_totp
```

Step III: Identification of high-order moments

- **Using knowable moments** (K-moments; Koutsoyiannis, 2019) for empirical values of intensities x:

$$\underline{\hat{K}'_p} = \sum_{i=1}^{n} b_{inp}\, \underline{x}_{(i:n)}$$

$$b_{inp} = \begin{cases} 0, & i < p \\ \dfrac{p}{n}\, \dfrac{\Gamma(n-p+1)}{\Gamma(n)}\, \dfrac{\Gamma(i)}{\Gamma(i-p+1)}, & i \geq p \geq 0 \end{cases}$$

where $\underline{x}_{(i:n)}$ is the ith element of a sample of $\underline{x}$ of size n, sorted in ascending order and p is the moment order which can be any positive number $\leq n$ (usually, but not necessarily, integer).

- **Example Python implementation:**

```
## K-moment for order p
def kmom(p,x): ## Non-central Unbiased for q=1, x must be sorted in increasing order
    x=x[~np.isnan(x)]
    n=len(x)
    b=np.empty(n)
    b[:]=np.NaN
    i=np.arange(1,n+1,1)
    i_0=i[i<p]
    i_1=i[i>=p]
    b[i_0-1]=0
    b[i_1-1]=(p/n)*np.exp(gammaln(n-p+1)-gammaln(n)+gammaln(i_1)-gammaln(i_1-p+1))
    kmom=np.sum(b*x)
    return kmom
```

- **More details in Section 6.9 (SoE)**

Step IV: Assigning return periods

- **Based on K-moments:**

$$T\left(\underline{\hat{K}}_p'\right) = \frac{k}{P_1^{(k)}} p' \Lambda_{p'} \approx \frac{k}{P_1^{(k)}} \left(\Lambda_\infty p' + (\Lambda_1 - \Lambda_\infty)\right)$$

- For Pareto scales

$$\Lambda_1 = (1-\xi)^{-\frac{1}{\xi}} \qquad \Lambda_\infty = \Gamma(1-\xi)^{\frac{1}{\xi}}$$

- For PBF scales

$$\Lambda_1 = \left(1 + \left(\frac{\mathrm{B}\left(\frac{1}{\zeta\xi} - \frac{1}{\zeta}, \frac{1}{\zeta}\right)}{\zeta}\right)^{\zeta}\right)^{\frac{1}{\zeta\xi}} \qquad \Lambda_\infty = \Gamma(1-\xi)^{\frac{1}{\xi}}$$

where p' the bias corrected moment order accounting for time dependence.

- **Example Python implementation:**

```
## K return periods
def K_return_periods(params,s_len,d_len,k_trans,k_h,scales_h,scales_d,Cg_type,pw_k,p):

...

    for i in np.arange(0,len(scales)):
        scale=scales[i]
        K_rt[scale][:]=(Linf[scale]*(p_2[scale].values-1)+L1_P[scale])*k_y[i]/pw_k[scale]
    return K_rt
```

- **More details in Section 6.14 (SoE)**

T. Iliopoulou; D. Koutsoyiannis

Step V: Calibration

- **Minimizing an error metric focusing on distribution quantiles $x(k,T)$ for all available time scales k and a series of return periods T :**

$$E_x \coloneqq \sum_k \frac{1}{\gamma(k)} \frac{1}{n_k} \sum_T w_x(T)\big(x(k,T) - \hat{x}(k,T)\big)^2$$

where $w_x(T)$ is a weighting factor as a function of the return period T, and n_k is the number of x values at time scale k.

Parameter	Meaning of parameter
μ	Mean intensity
λ_1, λ_2	Intensity scale parameters
α	Time scale parameter
M	Fractal (smoothness) parameter
H	Hurst parameter
θ	Exponent of the expression of probability dry
ξ	Tail index

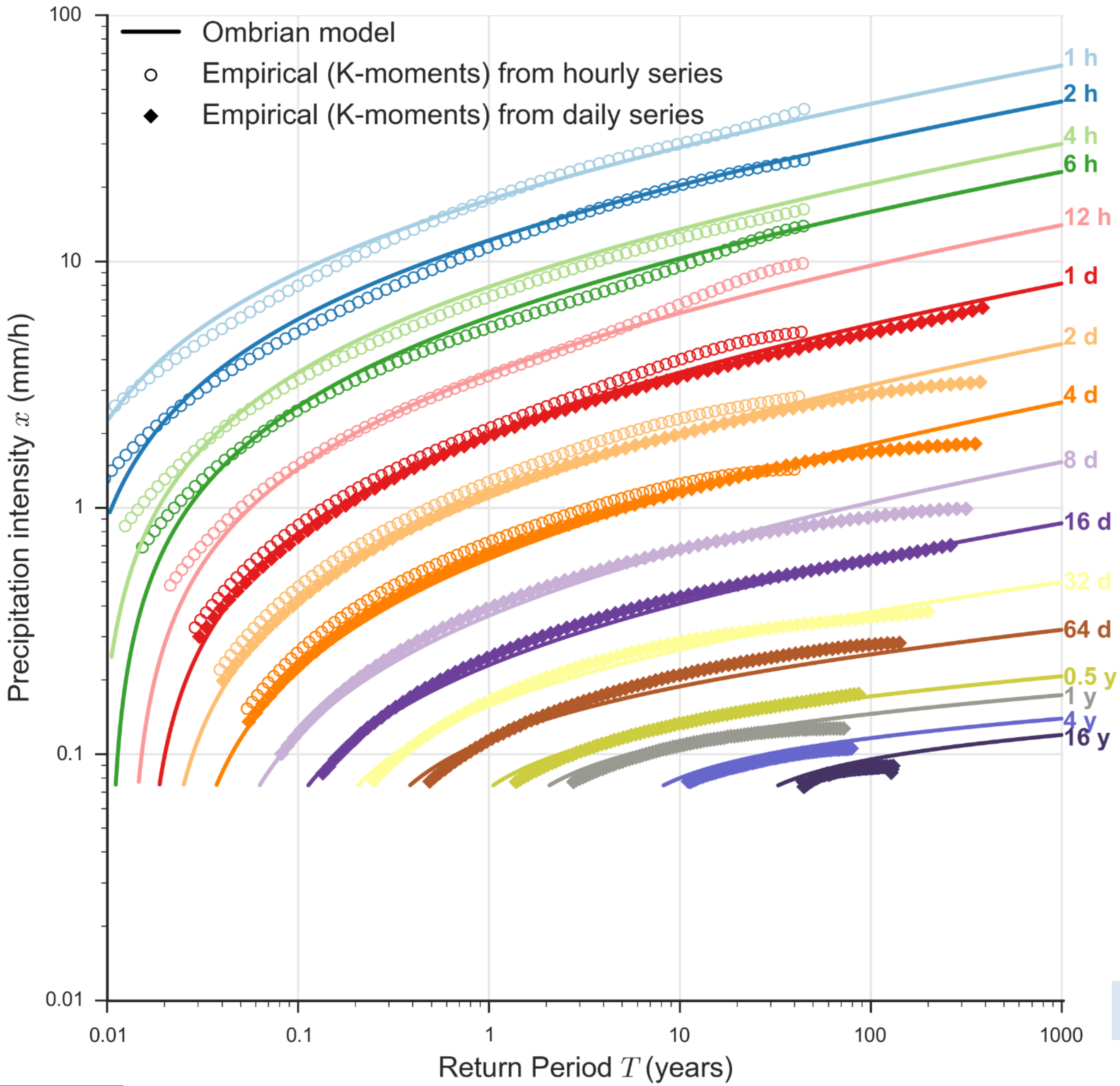
- **Example Python implementation:**

```
### Model error based on quantiles
def ombrian_quant_error(params,s_len,d_len,k_trans,k_h,scales_h,scales_d,Cg_type,pw_k,Ekmoms,p):
    scales=scales_h+scales_d
    l1=params[2]
    if Cg_type=='CD':
       l2=params[3]
       a=params[4]
       H=params[5]
...
        E_stand=E/th_var_k
        Etot0=E_stand.sum(axis=1)
        Etot=Etot0.values
    return Etot
```

Minimize using SciPy's global optimizers (e.g. differential evolution)

Application: 1 h to 16 y rainfall in Bologna

Parameterization of the model and visualization of the results



Using:

- Hourly series (1990-2013)
- Daily timeseries of Bologna (1813-2018) (Koutsoyiannis, 2021)

- Climacogram type: CD
- Transition time-scale: 96 h

Parameters

μ	0.0746
λ_1	0.0011
λ_2	2.1986
α	8.4341
H	0.95
θ	1
ξ	0.11067

➢ **More details in Digression 8.E (SoE)**

Overview of toolbox structure

Input data

Use full rainfall series at a sub-daily time-scale & optionally blend with other daily series

User choices

Choose range of time-scales for the calibration of the ombrian model, e.g. from the minimum available to years

Toolbox functions

Empirical climacogram ⟹ Choose climacogram model & obtain first guess of climacogram parameters

Empirical probability wet vs scale ⟹ Choose transition time-scale from Pareto to PBF distribution

Empirical *K*-moments

Minimize error between empirical *K*-moments and model quantiles

End result: Parameters of the ombrian model

Package dependencies: NumPy, Pandas, SciPy, Matplotlib, Seaborn

Summary

- Advancing empirically-derived ombrian curves to theoretically-consistent ombrian models allows the user to address bias and estimation uncertainty, extrapolate results to longer timescales and perform simulation for complex hydrological systems.
- The PythOm toolbox implements these advances in an easy and nearly-automated manner, requiring minimal choices by the user.
- The toolbox is currently in beta testing and will be released soon alongside user manual at https://www.itia.ntua.gr/en/docinfo/2111/.

References:

- D. Koutsoyiannis, *Stochastics of Hydroclimatic Extremes - A Cool Look at Risk, ISBN: 978-618-85370-0-2*, 333 pages, Kallipos, Athens, 2021. http://www.itia.ntua.gr/en/docinfo/2000/.
- D. Koutsoyiannis, An entropic-stochastic representation of rainfall intermittency: The origin of clustering and persistence, *Water Resources Research*, 42 (1), W01401, doi:10.1029/2005WR004175, 2006.
- D. Koutsoyiannis and T. Iliopoulou. Ombrian curves advanced to stochastic modelling of rainfall intensity, *Rainfall: modeling, measurement and applications*, edited by R. Morbidelli, Chapter 9, Elsevier, 2021 (*in press*).

Contact: T. Iliopoulou; tiliopoulou@hydro.ntua.gr, D. Koutsoyiannis; dk@itia.ntua.gr