OpenTURNS Users Day #17

Focus on

« The extreme values modelling capacities

of OpenTURNS v1.23 »

A. Dutfoy (EDF)
June 2024

Extreme values modelling capacities of OpenTURNS v1.23

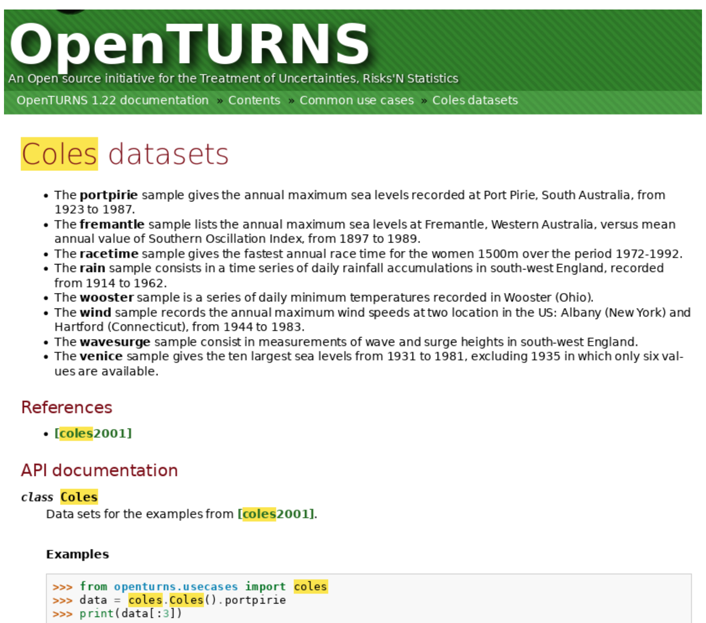
Both approaches are developped in the library:

- **Bloc maxima approach**: based on the **Generalized Extreme Value distribution (GEV)**
- **Peak Over Threshold approach:** based on the **Generalized Pareto Distribution (GPD)**

References:

- **Coles** : « An introduction to statistical modelling of extreme values », Springer 2001
- **Beirlant** : « Statistics of extremes: theory and application » , Wiley, 2004.

Examples: all the examples developped by Coles are reproduced in the example documentation of the library.


OpenTURNS
An Open source initiative for the Treatment of Uncertainties, Risks'N Statistics
OpenTURNS 1.22 documentation » Contents » Common use cases » Coles datasets

Coles datasets

- The **portpirie** sample gives the annual maximum sea levels recorded at Port Pirie, South Australia, from 1923 to 1987.
- The **fremantle** sample lists the annual maximum sea levels at Fremantle, Western Australia, versus mean annual value of Southern Oscillation Index, from 1897 to 1989.
- The **racetime** sample gives the fastest annual race time for the women 1500m over the period 1972-1992.
- The **rain** sample consists in a time series of daily rainfall accumulations in south-west England, recorded from 1914 to 1962.
- The **wooster** sample is a series of daily minimum temperatures recorded in Wooster (Ohio).
- The **wind** sample records the annual maximum wind speeds at two location in the US: Albany (New York) and Hartford (Connecticut), from 1944 to 1983.
- The **wavesurge** sample consist in measurements of wave and surge heights in south-west England.
- The **venice** sample gives the ten largest sea levels from 1931 to 1981, excluding 1935 in which only six values are available.

References

- [coles2001]

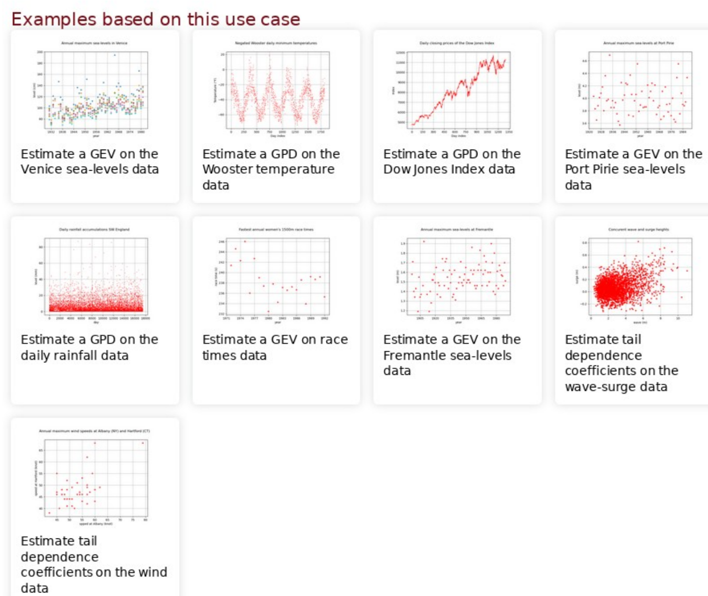
API documentation

class **Coles**

Data sets for the examples from [coles2001].

Examples

```
>>> from openturns.usecases import coles
>>> data = coles.Coles().portpirie
>>> print(data[:3])
```



Extreme values modelling capacities of OpenTURNS v1.23

Block Maxima approach: based on the **Generalized Extreme Value Distribution**

GeneralizedExtremeValueFactory

`build(*args)`	Estimate the distribution via maximum likelihood.
`buildAsGeneralizedExtremeValue(*args)`	Estimate the distribution as native distribution.
★ `buildCovariates(*args)`	Estimate a GEV from covariates.
`buildEstimator(*args)`	Build the distribution and the parameter distribution.
★ `buildMethodOfLikelihoodMaximization(sample)`	Estimate the distribution from the r largest order statistics.
`buildMethodOfLikelihoodMaximizationEstimator(sample)`	Estimate the distribution and the parameter distribution with the R-maxima method.
★ `buildMethodOfXiProfileLikelihood(sample[, r])`	Estimate the distribution with the profile likelihood.
`buildMethodOfXiProfileLikelihoodEstimator(sample)`	Estimate the distribution and the parameter distribution with the profile likelihood.
★ `buildReturnLevelEstimator(result, m)`	Estimate a return level and its distribution from the GEV parameters.
★ `buildReturnLevelProfileLikelihood(sample, m)`	Estimate a return level and its distribution with the profile likelihood.
`buildReturnLevelProfileLikelihoodEstimator(...)`	Estimate (z_m, σ, ξ) and its distribution with the profile likelihood.
★ `buildTimeVarying(*args)`	Estimate a non stationary GEV from a time-dependent parametric model.
`getBootstrapSize()`	Accessor to the bootstrap size.
`getClassName()`	Accessor to the object's name.
`getName()`	Accessor to the object's name.
`getOptimizationAlgorithm()`	Accessor to the solver.
`hasName()`	Test if the object is named.
`setBootstrapSize(bootstrapSize)`	Accessor to the bootstrap size.
`setName(name)`	Accessor to the object's name.
`setOptimizationAlgorithm(solver)`	Accessor to the solver.

❖ **Stationary GEV**

- **GEV estimate**:
  - Max log-likelihood based on the maxima or the r highest maxima
  - Profile likelihood wrt ξ
- **Model validation**: QQ-plot, PP-Plot, Return level graph, model and Empirical pdf
- **Return level estimate**:
  - Max log-likelihood
  - Profile likelihood wrt the return level

Extreme values modelling capacities of OpenTURNS v1.23

Block Maxima approach: based on the **Generalized Extreme Value Distribution (GEV)**

Non stationary GEV: dependence to covariates

$$Z_t \sim \mathrm{GEV}(\mu(t), \sigma(t), \xi(t)) \qquad Z_{\boldsymbol{y}} \sim \mathrm{GEV}(\mu(\boldsymbol{y}), \sigma(\boldsymbol{y}), \xi(\boldsymbol{y}))$$

$$Z_t \sim \mathrm{GPD}(\sigma(t), \xi(t), u) \qquad Z_{\boldsymbol{y}} \sim \mathrm{GPD}(\sigma(\boldsymbol{y}), \xi(\boldsymbol{y}), u)$$

$$\theta_q(t) = h_q\left(\sum_{i=1}^{d_{\theta_q}} \beta_i^{\theta_q} \varphi_i^{\theta_q}(\tau(t))\right) \qquad \theta_q(y_1^q, \dots, y_{d_q}^q) = h_q\left(\sum_{i=1}^{d_q} \beta_i^q y_i^q + \beta_{d_q+1}^q\right)$$

GeneralizedExtremeValueFactory

`build(*args)`	Estimate the distribution via maximum likelihood.
`buildAsGeneralizedExtremeValue(*args)`	Estimate the distribution as native distribution.
`buildCovariates(*args)`	Estimate a GEV from covariates.
`buildTimeVarying(*args)`	Estimate a non stationary GEV from a time-dependent parametric model.

- **Graphs t → θ(t) or y→ θ(y)**
- **Graphs t → $q_p(Z(t))$ or y → $q_p(Z(y))$**
- **Model validation (standardized distribution)**: QQ-plot, PP-Plot, Return level graph, model and Empirical pdf

TimeVaryingResult

`drawDiagnosticPlot()`	Draw the 4 usual diagnostic plots.
`drawParameterFunction([parameterIndex])`	Draw the parameter function.
`drawQuantileFunction(p)`	Draw the quantile function.
`getClassName()`	Accessor to the object's name.
`getDistribution(t)`	Accessor to the Parent distribution at a given time.
`getLogLikelihood()`	Optimal log-likelihood value accessor.
`getName()`	Accessor to the object's name.
`getNormalizationFunction()`	Normalizing function accessor.
`getOptimalParameter()`	Optimal parameter accessor.
`getParameterDistribution()`	Accessor to the distribution of β.
`getParameterFunction()`	Parameter function accessor.
`getTimeGrid()`	Accessor to the time grid.
`hasName()`	Test if the object is named.
`setLogLikelihood(logLikelihood)`	Optimal log-likelihood value accessor.
`setName(name)`	Accessor to the object's name.
`setParameterDistribution(parameterDistribution)`	Accessor to the distribution of of β.

CovariatesResult

`drawParameterFunction1D(*args)`	Draw the parameter function.
`drawParameterFunction2D(*args)`	Draw the parameter function.
`drawQuantileFunction1D(*args)`	Draw the quantile function.
`drawQuantileFunction2D(*args)`	Draw the quantile function.
`getClassName()`	Accessor to the object's name.
`getCovariates()`	Covariates accessor.
`getDistribution(covariates)`	Accessor to the Parent distribution at a given covariate vector.
`getLogLikelihood()`	Optimal likelihood value accessor.
`getName()`	Accessor to the object's name.
`getNormalizationFunction()`	Normalizing function accessor.
`getOptimalParameter()`	Optimal parameter accessor.
`getParameterDistribution()`	Accessor to the distribution of β.
`getParameterFunction()`	Parameter function accessor.
`hasName()`	Test if the object is named.
`setLogLikelihood(logLikelihood)`	Optimal likelihood value accessor.
`setName(name)`	Accessor to the object's name.
`setParameterDistribution(parameterDistribution)`	Accessor to the distribution of of β.

Model selection based on the Likelihood Ratio test

Hypothesis tests

`HypothesisTest.ChiSquared(firstSample, ...)`	Test whether two discrete samples are independent.
`HypothesisTest.LikelihoodRatioTest(...[, level])`	Nested likelihood model selection.

Extreme values modelling capacities of OpenTURNS v1.23

Block Maxima approach: based on the **Generalized Extreme Value Distribution (GEV)**

❖ **Model validation**

GeneralizedExtremeValueValidation

`drawDiagnosticPlot()`	Draw the 4 usual diagnostic plots.
`drawPDF()`	Draw the estimated density and the data histogram.
`drawReturnLevel()`	Draw the return level with confidence interval.
`getClassName()`	Accessor to the object's name.
`getConfidenceLevel()`	Confidence level accessor.
`getName()`	Accessor to the object's name.
`hasName()`	Test if the object is named.
`setConfidenceLevel(confidenceLevel)`	Confidence level accessor.
`setName(name)`	Accessor to the object's name.

- **Four usual graphs: :**
  - QQ-plot,
  - PP-Plot,
  - Return level graph,
  - model and Empirical pdf

Performed on standardized distributions in case of time or covariates depdence

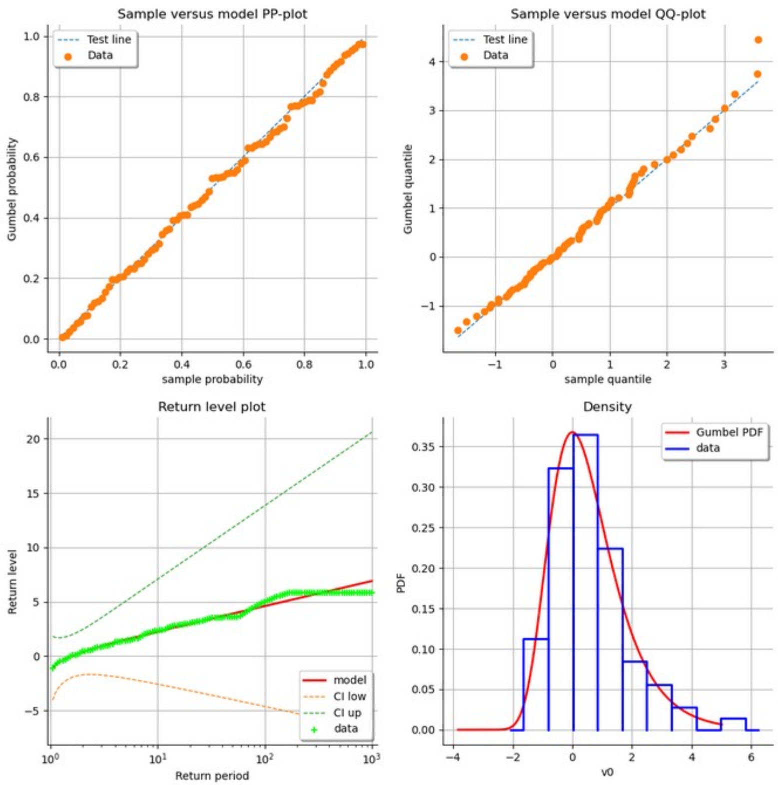
$$Z_t \sim \mathrm{GEV}(\mu(t), \sigma(t), \xi(t))$$
$$Z_t \sim \mathrm{GPD}(\sigma(t), \xi(t), u)$$

$$\hat{Z}_t = \frac{1}{\xi(t)} \log\left[1 + \xi(t)\left(\frac{Z_t - \mu(t)}{\sigma(t)}\right)\right]$$
$$\hat{Z}_t = \frac{1}{\xi(t)} \log\left[1 + \xi(t)\left(\frac{Z_t - u}{\sigma(t)}\right)\right]$$

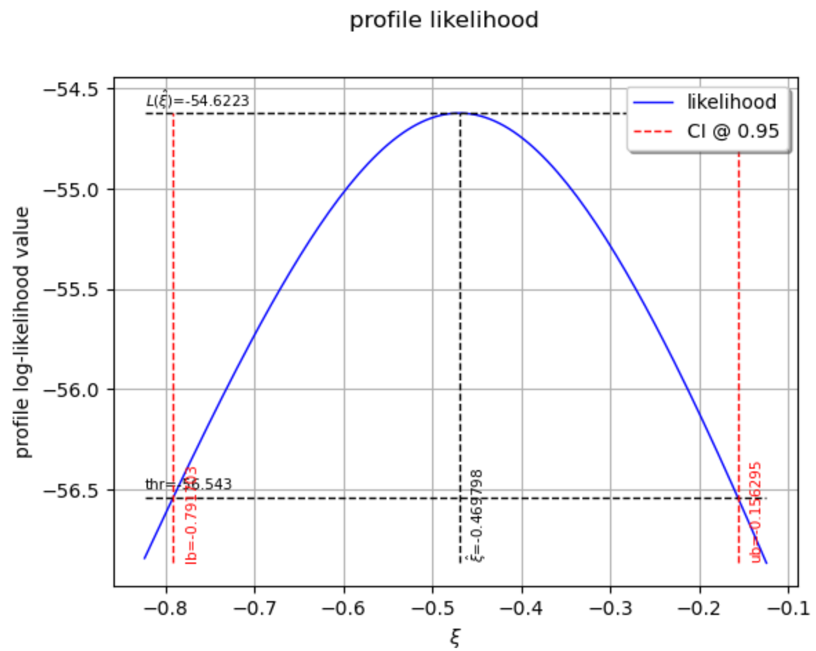
Extreme values modelling capacities of OpenTURNS v1.23

Block Maxima approach: based on the **Generalized Extreme Value Distribution (GEV)**

- Validation graphs



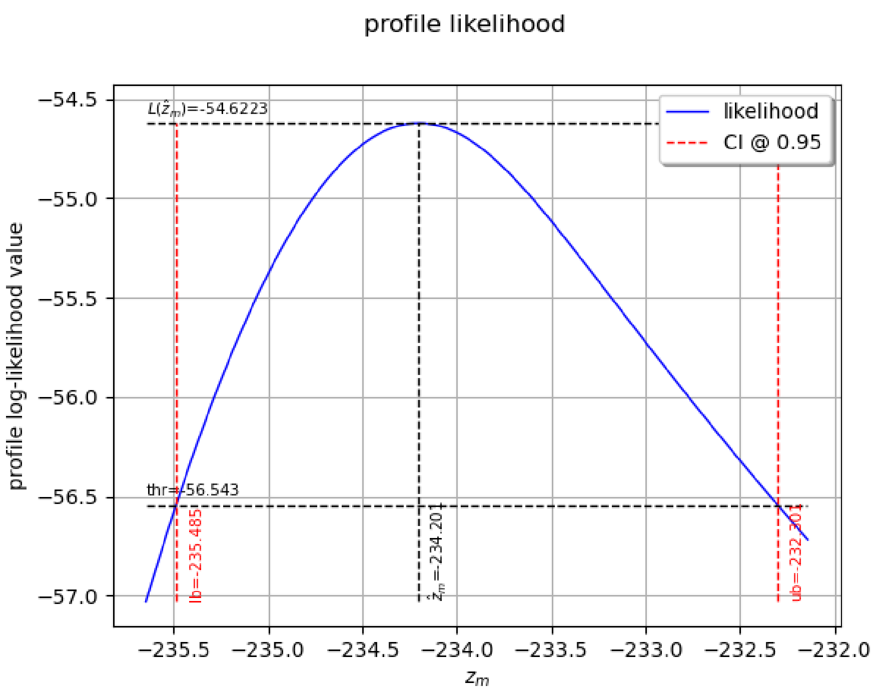
- ξ estimation from the profile likelihood



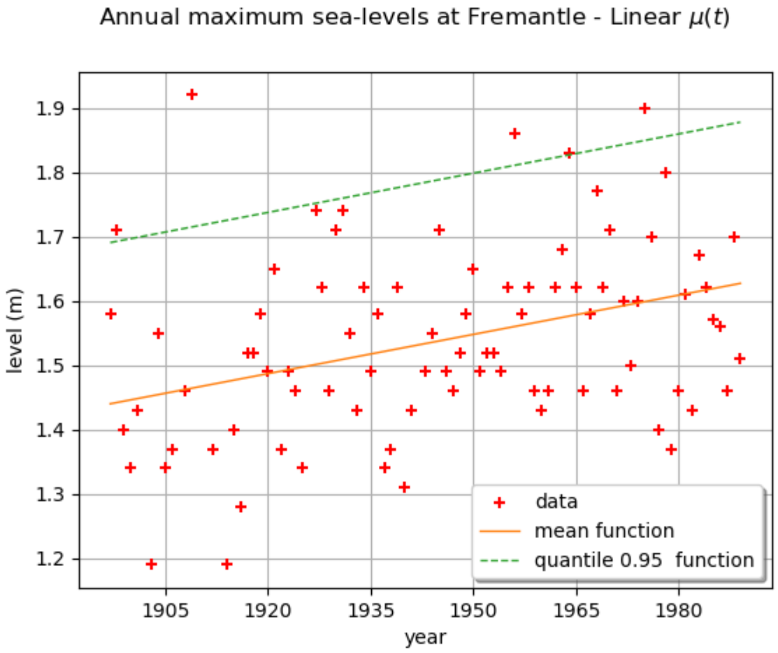
Extreme values modelling capacities of OpenTURNS v1.23

Block Maxima approach: based on the **Generalized Extreme Value Distribution (GEV)**

- Return level estimation from the profile likelihood



- t → µ(t) of Z(t)



Extreme values modelling capacities of OpenTURNS v1.23

Peak Over Threshold approach: based on the **Generalized Pareto Distribution (GPD)**

GeneralizedParetoFactory

`build`(*args)	Build the distribution.
`buildAsGeneralizedPareto`(*args)	Build the distribution as a GeneralizedPareto type.
`buildCovariates`(*args)	Estimate a GPD from covariates.
`buildEstimator`(*args)	Build the distribution and the parameter distribution.
`buildMethodOfExponentialRegression`(sample)	Build the distribution based on the exponential regression estimator.
`buildMethodOfLikelihoodMaximization`(sample, u)	Estimate the distribution with the maximum likelihood method.
`buildMethodOfLikelihoodMaximizationEstimator`(...)	Estimate the distribution and the parameter distribution with the maximum likelihood method.
`buildMethodOfMoments`(sample)	Build the distribution based on the method of moments estimator.
`buildMethodOfProbabilityWeightedMoments`(sample)	Build the distribution based on the probability weighted moments estimator.
`buildMethodOfXiProfileLikelihood`(sample, u)	Estimate the distribution with the profile likelihood.
`buildMethodOfXiProfileLikelihoodEstimator`(...)	Estimate the distribution and the parameter distribution with the profile likelihood.
`buildReturnLevelEstimator`(result, sample, m)	Estimate a return level and its distribution from the GPD parameters.
`buildReturnLevelProfileLikelihood`(sample, u, m)	Estimate a return level and its distribution with the profile likelihood.
`buildReturnLevelProfileLikelihoodEstimator`(...)	Estimate (z_m, ξ) and its distribution with the profile likelihood.
`buildTimeVarying`(*args)	Estimate a non stationary GPD from a time-dependent parametric model.
`drawMeanResidualLife`(sample)	Draw the mean residual life plot.
`drawParameterThresholdStability`(sample, ...)	Draw the parameter threshold stability plot.
`getBootstrapSize`()	Accessor to the bootstrap size.
`getClassName`()	Accessor to the object's name.
`getName`()	Accessor to the object's name.
`getOptimizationAlgorithm`()	Accessor to the solver.
`hasName`()	Test if the object is named.
`setBootstrapSize`(bootstrapSize)	Accessor to the bootstrap size.
`setName`(name)	Accessor to the object's name.
`setOptimizationAlgorithm`(solver)	Accessor to the solver.

❖ Stationary GPD

- **Threshold selection**: Mean Residual Life Plot, Parameter Threshold Stability
- **GPD estimate**:
  - Max log-likelihood
  - Profile likelihood wrt ξ
- **Model validation**: QQ-plot, PP-Plot, Return level graph, model and Empirical pdf
- **Return level estimate**:
  - Max log-likelihood
  - Profile likelihood wrt the return level

Extreme values modelling capacities of OpenTURNS v1.23

Peak Over Threshold approach: based on the **Generalized Pareto Distribution (GPD)**

SamplePartition

`ExtractFromDataFrame`(partial)	Extract a partition from a pandas dataframe as a SamplePartition.
`draw`(threshold)	Draw clusters and peaks.
`getClassName`()	Accessor to the object's name.
`getIndicesCollection`()	Partition indices accessor.
`getName`()	Accessor to the object's name.
`getPeakOverThreshold`(threshold, r)	Compute extreme values using Peaks Over Threshold (POT) method.
`getSample`()	Sample accessor.
`hasName`()	Test if the object is named.
`setName`(name)	Accessor to the object's name.

❖ **Non stationary GPD: select stationary periods and independent data**

- **Stationary period selection**: use of **pandas**
- **Clusters and peak selection**

❖ **Non stationary GPD: dependence to covariates**

$$Z_t \sim \mathrm{GEV}(\mu(t), \sigma(t), \xi(t)) \qquad Z_{\boldsymbol{y}} \sim \mathrm{GEV}(\mu(\boldsymbol{y}), \sigma(\boldsymbol{y}), \xi(\boldsymbol{y}))$$

$$Z_t \sim \mathrm{GPD}(\sigma(t), \xi(t), u) \qquad Z_{\boldsymbol{y}} \sim \mathrm{GPD}(\sigma(\boldsymbol{y}), \xi(\boldsymbol{y}), u)$$

$$\theta_q(t) = h_q\left(\sum_{i=1}^{d_{\theta_q}} \beta_i^{\theta_q} \varphi_i^{\theta_q}(\tau(t))\right) \qquad \theta_q(y_1^q, \dots, y_{d_q}^q) = h_q\left(\sum_{i=1}^{d_q} \beta_i^q y_i^q + \beta_{d_q+1}^q\right)$$

GeneralizedParetoFactory

`build`(*args)	Build the distribution.
`buildAsGeneralizedPareto`(*args)	Build the distribution as a GeneralizedPareto type.
`buildCovariates`(*args)	Estimate a GPD from covariates.
`buildTimeVarying`(*args)	Estimate a non stationary GPD from a time-dependent parametric model.

- **Graphs t → θ(t) or y→ θ(y)**
- **Graphs t → $q_p(Z(t))$ or y → $q_p(Z(y))$**

Extreme values modelling capacities of OpenTURNS v1.23

Peak Over Threshold approach: based on the **Generalized Pareto Distribution (GPD)**

❖ **Model validation**

GeneralizedParetoValidation

`drawDiagnosticPlot()`	Draw the 4 usual diagnostic plots.
`drawPDF()`	Draw the estimated density and the data histogram.
`drawReturnLevel()`	Draw the return level with confidence interval.
`getClassName()`	Accessor to the object's name.
`getConfidenceLevel()`	Confidence level accessor.
`getName()`	Accessor to the object's name.
`hasName()`	Test if the object is named.
`setConfidenceLevel(confidenceLevel)`	Confidence level accessor.
`setName(name)`	Accessor to the object's name.

- **Four usual graphs: :**
  - QQ-plot,
  - PP-Plot,
  - Return level graph,
  - model and Empirical pdf

Performed on standardized distributions in case of time or covariates depdence

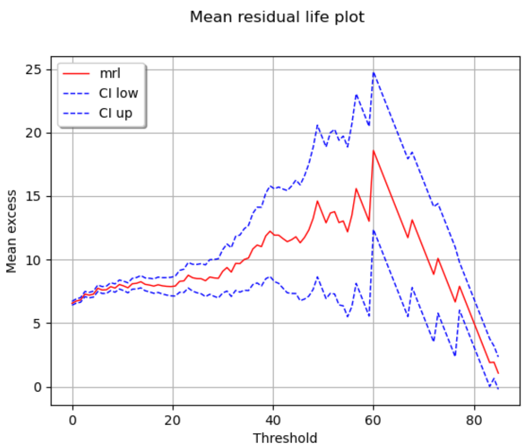
$$Z_t \sim \mathrm{GEV}(\mu(t), \sigma(t), \xi(t))$$
$$Z_t \sim \mathrm{GPD}(\sigma(t), \xi(t), u)$$

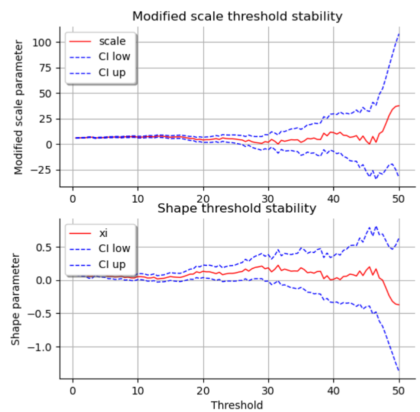
$$\hat{Z}_t = \frac{1}{\xi(t)} \log\left[1 + \xi(t)\left(\frac{Z_t - \mu(t)}{\sigma(t)}\right)\right]$$
$$\hat{Z}_t = \frac{1}{\xi(t)} \log\left[1 + \xi(t)\left(\frac{Z_t - u}{\sigma(t)}\right)\right]$$

Extreme values modelling capacities of OpenTURNS v1.23

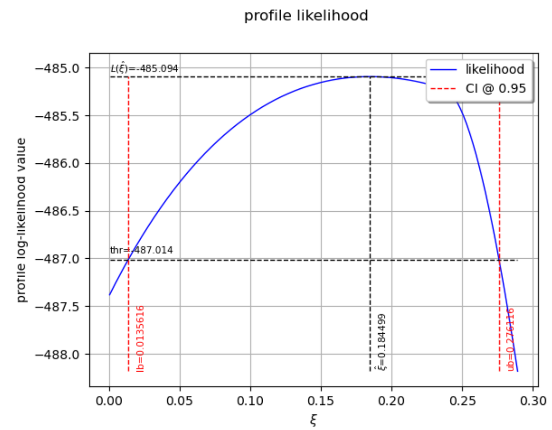
Peak Over Threshold approach: based on the **Generalized Pareto Distribution (GPD)**

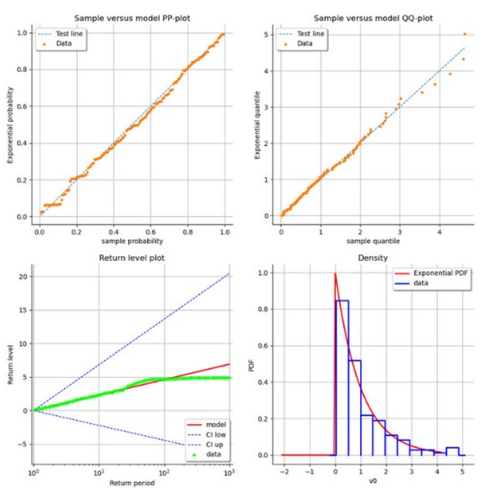
- Threshold selection: mean residual life plot and modified parameters graph





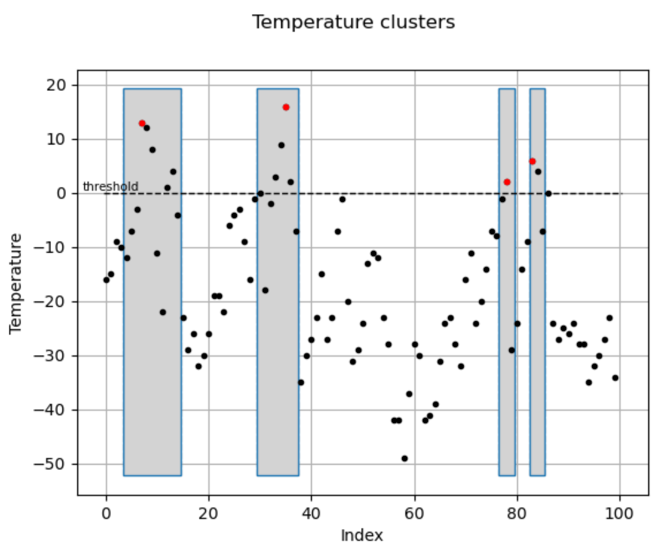
- ξ estimation from the profile likelihood





- Validation graphs

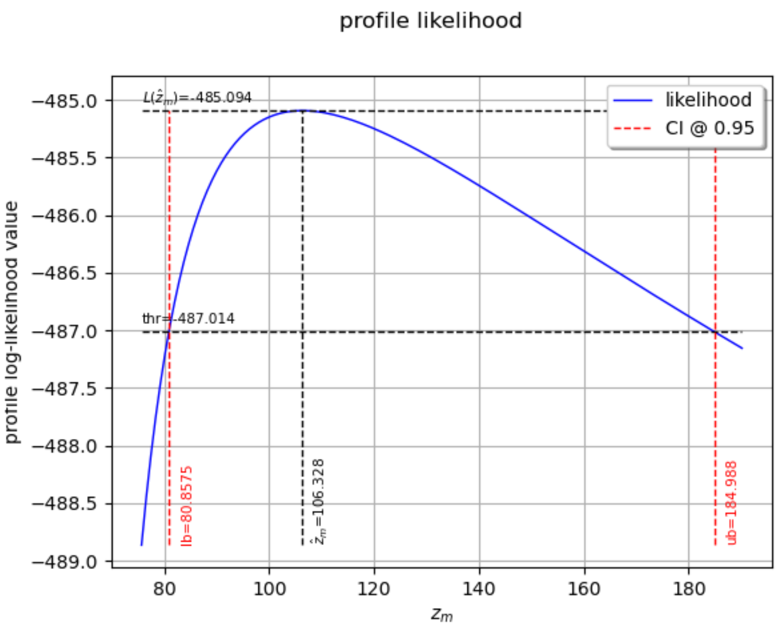
- Clusters, Peaks over threshold



Extreme values modelling capacities of OpenTURNS v1.23

Peak Over Threshold approach: based on the **Generalized Pareto Distribution (GPD)**

- Return level estimation from the profile likelihood



- t → σ(t) of Z(t)