Requirements for the Module statistics_lib.inverse_distributions

Conventions

Requirements listed in this document are constructed according to the following structure:

Requirement ID: REQ-UVW-XYZ

Title: Title / name of the requirement

Description: Description / definition of the requirement

Verification Method: I / A / T / D

The requirement ID starts with the fixed prefix 'REQ'. The prefix is followed by 3 letters abbreviation (in here 'UVW'), which defines the requiement type - e.g. 'FUN' for a functional and capability requirement, 'AWM' for an alarm, warnings and operator messages, etc. The last part of the ID is a 3-digits *hexadecimal* number (0..9|A..F), with the first digit identifing the module, the second digit identifing a class / function, and the last digit - the requirement ordering number for this object. E.g. 'REQ-FUN-112'. Each requirement type has its own counter, thus 'REQ-FUN-112' and 'REQ-AWN-112' requirements are different entities, but they refer to the same object (class or function) within the same module.

The verification method for a requirement is given by a single letter according to the table below:

Term	Definition
Inspection (I)	Control or visual verification
Analysis (A)	Verification based upon analytical evidences
Test (T)	Verification of quantitative characteristics with quantitative measurement
Demonstration (D)	Verification of operational characteristics without quantitative measurement

Functional and capability requirements

Requirement ID: REQ-FUN-600

Title: Functionality implemented by the module (scope)

Description: The module should provide classes implementing the following continuous random distribution using the definitions in DE004 document:

- Inverse Gaussian distribution
- Inverse Gamma distribution
- Inverse χ^2-distribution
- Scaled inverse χ^2-distribution
- Cauchy distribution
- Levy distribution

These functionality is additional, and it is not a part of the overal library requirements. However, the functionality of these classes must conform the same requirements as of the mandatory distribution classes, and these classes must have the same API.

Verification Method: A

Requirement ID: REQ-FUN-601

Title: Instantiation of a random distribution class

Description: All classes should accept and require as many aruments of the initialization method (during instantiation) as they have parameters, defining the distribition. These arguments must be of the proper data type and of the accepted values range, as the specific distribution is defined for.

Verification Method: T

Requirement ID: REQ-FUN-602

Title: Access to the parameters of a random distribution

Description: All parameters of the distribution should be accessible via a respective getter property. It should be possible to change the value of each parameter via a respective setter property, as long as the assigment value is of the proper type and value.

Verification Method: T

Requirement ID: REQ-FUN-603

Title: Required statistical properties

Description: All distribution classes should provide the following read-only properties representing the statistical properties of the distribution:

- Mean
- Max
- Min
- Median
- Q1
- Q3
- Var
- Sigma
- Skew
- Kurt

The exception is inverse Gaussian distribution, for which Mean is also the parameter of the distribution, therefore it must be read-write property. These properties should have floating point type or **None** value (in not defined), which values match the definitions in DE004 document.

Verification Method: T

Requirement ID: REQ-FUN-604

Title: Probability density function

Description: All classes should have instance method *pdf*(), which returns the value of the probability density function for a continuous distribution if the passed argument is an integer or floating point number (real) and of the value for which the distribution is defined. In the case of the real number argument with the value outside the acceptable range - zero value (0) should be returned instead of rising an exception.

Verification Method: T

Requirement ID: REQ-FUN-605

Title: Cummulative probability function

Description: All classes should have instance method *cdf*(), which returns the value of the cummulative probability density function. For a continuous distribution it should be a continuous function. It should accept any real number value as its argument. If it is less than the minimum of the aceeptance range the returned value should be zero (0), and one (1) if the value is greater than the maximum of the acceptance range.

Verification Method: T

Requirement ID: REQ-FUN-606

Title: Quantile function

Description: All classes should have instance method *qf*(), which returns the value of the quantile function, which is inverse to the cummulative probability density function. It should accept only floating point numbers in the open range (0, 1). It should return a floating point number within the range for which the distribution is defined. The second function *getQuantile*(k, m) should be defined for 0 < k : **int** < m : **int** and return *qf*(k/m).

Verification Method: T

Requirement ID: REQ-FUN-607

Title: Generation of histogram of distribution

Description: All classes should have instance method *getHistogram*(), which should calculate a histrogram of the distribution as follows:

- the arguments of the method are: central value of the minimal bin X_{min}, central value of the maximal bin X_{max} and the number of bins $N > 1$
- each bin has the same width $S = \frac{X_{max} - X_{min}}{N-1}$
- the bins are indexed from 0, and the central value of each bin is $x_k = X_{min} + k * S$, where $0 \leq k \leq N-1$
- for the k-th bin the value is calculated as $cdf(x_k + S/2) - cdf(x_k - S/2)$

- the result is returned as a tuple of 2-tuples with each nested tuple being a pair of the central value and the frequency for the given bin, where the bins are sorted in the ascending order of the central values

Verification Method: D

Requirement ID: REQ-FUN-608

Title: Generation of a random number

Description: All classes should have instance method w/o arguments *random*(), which returns a random value (floating point), with those numbers being distributed with the respective random disribution.

Verification Method: D

Alarms, warnings and operator messages

Requirement ID: REQ-AWM-600

Title: Improper type of an agrument

Description: The **TypeError** or its sub-class should be raised if an improper data type argument is passed into any parametric method of any class, including the initialization / instantiation method, and assigment to a setter property. Specifically,

- Instantiation of the classes and changing the distribution parameters via setter properties - any of the parameters is not a real number (**int** or **float** type)
- pdf() and cdf() - the argument is not **int** or **float**
- qf() - the argument is not **float**
- getQuantile(k, m) - either of the arguments is not **int**
- getHistogram(min, max, NBins) - either min or max is not **int** or **float**, or NBins is not **int**

Verification Method: T

Requirement ID: REQ-AWM-601

Title: Improper value of an agrument

Description: The **ValueError** or its sub-class should be raised if an argument of a proper data type, but of unacceptable value is passed into any parametric method of any class, including the initialization / instantiation method, and assigment to a setter property. Specifically,

- Instantiation of the classes and changing the distribution parameters via setter properties
    - Levy and Cauchy distributions - scale parameter is not positive, i.e. <= 0
    - Other distributions - any of the parameters is not positive, i.e. <= 0
- qf() - the argument is not within (0, 1) range
- getQuantile(k, m) - either of the arguments is <= 0, or k >= m
- getHistogram(min, max, NBins) - min >= max , or NBins is <= 1

Verification Method: T