# BloxMath Library Reference

*Release 3.9*

**LogicBlox**

April 25, 2012

# CONTENTS

CHAPTER

ONE

# INTRODUCTION

The library **BloxMath** encompasses a collection of mathematical, and other functions that are made available to the user as built-in predicates. Much of the functionality of mathematical libraries such as Boost.Math are thus made available in LogicBlox.

- Statistical distribution functions
- Financial formatting functions
- Various mathematical functions
- Linguistic processing functions
- Random number generation
- Optimization of smoothing

## 1.1 Using The Library

None of the functionality from BloxMath is available to the user without loading the BloxMath library. In bloxbatch or bloxlib this would be accomplised by adding the `-lib BloxMath` parameter to the executable when a workspace is created or opened.

CHAPTER

TWO

# FINANCIAL FORMATTING FUNCTIONS

The financial formatting functions comprise a group of built-in functions that take a floating numerical value and convert it to a string representing some particular convention of displaying the numerical value as currency. Consider a simple query below:

```
_[]=v <- float64:financial:USD:string[-12345.678]=v.
```

The result is printed as the string `-$12,345.68`.

On the other hand, using the function `float:financial:EUR:string` will produce a string formatted for displaying Euros:

```
_[]=v <- float64:financial:EUR:string[-12345.678]=v.
```

The above code will print produce the string `-12.345,68 €`.

For a full list of financial functions, look in the index under the headings ''financial.'‘

CHAPTER

THREE

# STATISTICAL DISTRIBUTION FUNCTIONS

A comprehensive set of statistical distribution functions is provided. These can be used to build statistical tests and/or efficiently compute other quantities. The following distributions are covered: Normal (Gaussian) distribution, Student's T distribution, Chi squared distribution, Exponential distribution, Poisson distribution, Bernoulli distribution, Beta distribution, Gamma distribution, Binomial distribution, Negative binomial distribution, Laplace distribution, Uniform distribution, Fisher F distribution, Weibull distribution, Pareto distribution, Logistic distribution, Lognormal distribution, and Extreme value distribution.

For most distributions (i.e., those for which such functions are defined), the BloxMath library provides a set of built-in functions that compute the various statistical values of the distribution such as pdf (probability density function), cdf (cumulative density function), quantile, mean and standard deviation, mode, skewness, and kurtosis. Where such values are undefined, the functions may return `nan` or not be defined.

The statistical functions follow a certain pattern in naming. We will illustrate this on the example of the quantile function for a normal distribution. The function name is `float64:distribution:normal:quantile`. The name has several parts: first the type `float64` is followed by the word "distribution" which is followed by the name of the distribution (`normal`) which is followed by the name of the function (`quantile.`). Another function identical except in the first part (`float32`) is available for the single precision floating point numbers. The key space of the function (see index normal, quantile for the description of the function) is as follows: `float[64],float[64],float[64]`, i.e., the function takes three double precision arguments. In general the first n arguments are the the parameters of the distribution, followed by however many arguments the actual function needs. In this case, the first two are the mean and standard deviation that are used to define the particular normal distribution, followed by a number between 0 and 1 that indicates the quantile.

Compare the above function with `float64:distribution:rayleigh:quantile` which only takes two parameters. This function takes two parameter, the first of which is the sigma parameter of the Rayleigh distribution itself (as upposed to the two parameters needed to define a normal distribution), followed by the quantile argument between 0 and 1.

Most statistical functions in the library have a closed form solution that could be expressed in terms of more fundamental primitive functions already available in Datalog. However, the functions given in BloxMath should be used because they have improved numerical stability, accuracy and efficiency.

Another important thing to note is the binding pattern of the statistical functions: the functions in this library *always* require all their keys to be bound to particular value, and always themselves bind the value (i.e., there is no lookup of mean and/or standard deviation from the result of cdf for the normal distribution). It is up to the user to ensure that this invariant is followed, or the system will reject the datalog rule as not providing sufficient variable bindings.

To look up a particular function, look for either the name of the function (e.g., cdf) or the distribution (e.g., Pareto) in the index. If using a html or pdf version of this documentation, the index will have a link to short descriptions of the function in question.

## 3.1 Normal Distribution

Normal distribution (or Gaussian) is a continuous probability distribution that has a bell-shaped probability density function $f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2*\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ where $\mu$ is the mean and $\sigma^2$ is the variance. $\sigma$ is known as the standard deviation and is the argument that is actually expected by the BloxMath functions.

## 3.2 Student's t-distribution

Student's t-distribution (or simply the t-distribution) is a family of continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the sample size is small and population standard deviation is unknown.

The probability density function is parameterized by a parameter $\nu$, known as the degrees of freedom, and can be written as: $f(t) = \frac{1}{\sqrt{\nu}B(\frac{1}{2},\frac{\nu}{2})}(1+\frac{t^2}{\nu})^{-\frac{\nu+1}{2}}$ where B is the beta function. In addition to standard distribution-based functions (e.g., cdf, pdf, mean), this distribution has the function *float64:distribution:students_t:findDegreesOfFreedom* which estimates the degrees of freedom parameter given four arguments: diffference from the mean, alpha, beta and the standard deviation.

CHAPTER

FOUR

# SERIES FUNCTIONS

## 4.1 Random number generators

The BloxMath library provides a number of functions, part of the `series` predicate-to-predicate mapping, that allow the user to generate a collection of random numbers drawn from a particular distributions. The example below shows a simple usage:

```
s3[st,w]=v -> store(st), week(w), float[64](v).
s3[st,w]=v <- series<<v=rnd_bernoulli_distribution[w](m,seed)>>
   week(w), store(st), m = store:med[st], seed = store:number[st].
```

The predicate `s3` contains a set of random numbers drawn from a bermoulli distribution with median store:med[*st*] for each store `st`. All numbers along the time series dimension `[w]` will be drawn form the same distribution. In general the series function has the form `v = rnd_distrname_type(x1,...,xn,s)` where `x1` through `xn` are parameter values particular to the distribution in question, and s is an integer representing the random generator seed. The use of the seed is *mandatory*. The variables `x1`,..., `xn`, `s` must be bound on the right hand side of the `series<<...>>` rule.

- `rnd_uniform_int`
- `rnd_uniform_real`
- `rnd_bernoulli_distribution`
- `rnd_binomial_distribution`
- `rnd_cauchy_distribution`
- `rnd_gamma_distribution`
- `rnd_poisson_distribution`
- `rnd_geometric_distribution`
- `rnd_triangle_distribution`
- `rnd_exponential_distribution`
- `rnd_normal_distribution`
- `rnd_lognormal_distribution`

## 4.2 Multinomial distribution

Given a set of probabilities of distinct events with probabilities `probability[t]` adding up to 1.0, and a total number $n$ of trials, produces a result of a random series of $n$ outcomes, items[t]=v -> where `v` s add up to $n$. Example follows:

```
t(x), t:id(x:y) -> uint[32](y).
seed[]=v -> int[32](v).

probabilities[t]=v -> t(t), float[64](v).
items[t]=v -> t(t), int[32](v).
quantity[]=v -> uint[32](v).
quantity[]=1000.
items[t]=v <- series<<v=rnd_multinomial[t](p,n,seed)>> p = probabilities[t],
                                                       n = quantity[],
                                                       seed = seed[].
```

## 4.3 Optimized Smoothing

TBD

CHAPTER

FIVE

# REFERENCES

**float64:distribution:students_t:cdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:pdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:cdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:pdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:quantile**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for thestudents_t distribution.
The last argument must be 0 < x <= 1.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:chf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the students_t distribution:
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:variance**

*Keyspace* : float[64] *Valuespace* : float[64]
The variance for the students_t distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:mean**

*Keyspace* : float[64] *Valuespace* : float[64]
The mean for the students_t distribution. The mean function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float64:distribution:students_t:median**

*Keyspace* : float[64] *Valuespace* : float[64]
The median for the students_t distribution. The median function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**`float64:distribution:students_t:standardDeviation`**

*Keyspace* : float[64] *Valuespace* : float[64]
The standard deviation for students_t distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**`float64:distribution:students_t:kurtosis`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the students_t distribution. The kurtosis function no parameters other than ones required to construct the distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**`float64:distribution:students_t:kurtosisExcess`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the students_t distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**`float64:distribution:chi_squared:cdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**`float64:distribution:chi_squared:pdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:cdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:pdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:quantile**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for thechi_squared distribution.
The last argument must be 0 < x <= 1.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:chf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the chi_squared distribution:
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:variance**

*Keyspace* : float[64] *Valuespace* : float[64]

The variance for the chi_squared distribution.

The variance function no parameters other than ones required to construct the distribution (see above).

A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.

The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:mean**

*Keyspace* : float[64] *Valuespace* : float[64]

The mean for the chi_squared distribution. The mean function no parameters other than ones required to construct the distribution (see above).

A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.

The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:median**

*Keyspace* : float[64] *Valuespace* : float[64]

The median for the chi_squared distribution. The median function no parameters other than ones required to construct the distribution (see above).

A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.

The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float64:distribution:chi_squared:standardDeviation**

*Keyspace* : float[64] *Valuespace* : float[64]

The standard deviation for chi_squared distribution.

The standard deviation function no parameters other than ones required to construct the distribution (see above).

A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.

The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**`float64:distribution:chi_squared:kurtosis`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the chi_squared distribution. The kurtosis function no parameters other than ones required to construct the distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**`float64:distribution:chi_squared:kurtosisExcess`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the chi_squared distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**`float64:distribution:exponential:cdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:pdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:cdf:compl`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:pdf:compl`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:quantile`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for theexponential distribution.
The last argument must be 0 < x <= 1.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:chf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the exponential distribution:
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:variance`**

*Keyspace* : float[64] *Valuespace* : float[64]
The variance for the exponential distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:mean`**

*Keyspace* : float[64] *Valuespace* : float[64]
The mean for the exponential distribution. The mean function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:median`**

*Keyspace* : float[64] *Valuespace* : float[64]
The median for the exponential distribution. The median function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:standardDeviation`**

*Keyspace* : float[64] *Valuespace* : float[64]
The standard deviation for exponential distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:kurtosis`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the exponential distribution. The kurtosis function no parameters other than ones required to construct the distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:exponential:kurtosisExcess`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the exponential distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**`float64:distribution:rayleigh:cdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float64:distribution:rayleigh:pdf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:cdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:pdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:quantile**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for therayleigh distribution.
The last argument must be 0 < x <= 1.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:chf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the rayleigh distribution:
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:variance**

*Keyspace* : float[64] *Valuespace* : float[64]
The variance for the rayleigh distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float64:distribution:rayleigh:mean`**

*Keyspace* : float[64] *Valuespace* : float[64]
The mean for the rayleigh distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float64:distribution:rayleigh:median`**

*Keyspace* : float[64] *Valuespace* : float[64]
The median for the rayleigh distribution. The median function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float64:distribution:rayleigh:standardDeviation`**

*Keyspace* : float[64] *Valuespace* : float[64]
The standard deviation for rayleigh distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float64:distribution:rayleigh:kurtosis`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the rayleigh distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:rayleigh:kurtosisExcess**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the rayleigh distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float64:distribution:poisson:cdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:pdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:cdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:pdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:quantile**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for thepoisson distribution.
The last argument must be 0 < x <= 1.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:chf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the poisson distribution:
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:variance**

*Keyspace* : float[64] *Valuespace* : float[64]
The variance for the poisson distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:mean**

*Keyspace* : float[64] *Valuespace* : float[64]
The mean for the poisson distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:median**

*Keyspace* : float[64] *Valuespace* : float[64]
The median for the poisson distribution. The median function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:standardDeviation**

*Keyspace* : float[64] *Valuespace* : float[64]
The standard deviation for poisson distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k; \lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:kurtosis**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the poisson distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k; \lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:poisson:kurtosisExcess**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the poisson distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k; \lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float64:distribution:bernoulli:cdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the bernoulli distribution.

**float64:distribution:bernoulli:pdf**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The probability density function function for the bernoulli distribution.

**float64:distribution:bernoulli:cdf:compl**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the bernoulli distribution.

**`float64:distribution:bernoulli:pdf:compl`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the bernoulli distribution.

**`float64:distribution:bernoulli:quantile`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The quantile function for thebernoulli distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:bernoulli:chf`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The chf function for the the bernoulli distribution:

**`float64:distribution:bernoulli:variance`**

*Keyspace* : float[64] *Valuespace* : float[64]
The variance for the bernoulli distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:bernoulli:mean`**

*Keyspace* : float[64] *Valuespace* : float[64]
The mean for the bernoulli distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:bernoulli:median`**

*Keyspace* : float[64] *Valuespace* : float[64]
The median for the bernoulli distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:bernoulli:standardDeviation`**

*Keyspace* : float[64] *Valuespace* : float[64]
The standard deviation for bernoulli distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:bernoulli:kurtosis`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis for the bernoulli distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:bernoulli:kurtosisExcess`**

*Keyspace* : float[64] *Valuespace* : float[64]
The kurtosis excess for the bernoulli distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:normal:cdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the normal distribution.

**`float64:distribution:normal:pdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the normal distribution.

**`float64:distribution:normal:cdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the normal distribution.

**`float64:distribution:normal:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the normal distribution.

**`float64:distribution:normal:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thenormal distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:normal:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the normal distribution:

**`float64:distribution:normal:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the normal distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:normal:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the normal distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:normal:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the normal distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:normal:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for normal distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:normal:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the normal distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:normal:kurtosisExcess`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the normal distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:beta:cdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the beta distribution.

**`float64:distribution:beta:pdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the beta distribution.

**`float64:distribution:beta:cdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the beta distribution.

**`float64:distribution:beta:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the beta distribution.

**`float64:distribution:beta:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thebeta distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:beta:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the beta distribution:

**`float64:distribution:beta:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the beta distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:beta:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the beta distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:beta:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the beta distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:beta:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for beta distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:beta:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the beta distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:beta:kurtosisExcess`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the beta distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:gamma:cdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the gamma distribution.

**`float64:distribution:gamma:pdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the gamma distribution.

**`float64:distribution:gamma:cdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the gamma distribution.

**`float64:distribution:gamma:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the gamma distribution.

**`float64:distribution:gamma:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thegamma distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:gamma:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the gamma distribution:

**`float64:distribution:gamma:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the gamma distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:gamma:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the gamma distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:gamma:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the gamma distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:gamma:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for gamma distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:gamma:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the gamma distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:gamma:kurtosisExcess`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the gamma distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:binomial:cdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the binomial distribution.

**`float64:distribution:binomial:pdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the binomial distribution.

**`float64:distribution:binomial:cdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the binomial distribution.

**`float64:distribution:binomial:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the binomial distribution.

**`float64:distribution:binomial:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thebinomial distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:binomial:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the binomial distribution:

**`float64:distribution:binomial:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the binomial distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:binomial:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the binomial distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:binomial:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the binomial distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:binomial:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for binomial distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:binomial:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the binomial distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float64:distribution:binomial:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the binomial distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:negative_binomial:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the negative_binomial distribution.

**float64:distribution:negative_binomial:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the negative_binomial distribution.

**float64:distribution:negative_binomial:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the negative_binomial distribution.

**float64:distribution:negative_binomial:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the negative_binomial distribution.

**float64:distribution:negative_binomial:quantile**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thenegative_binomial distribution.
The last argument must be 0 < x <= 1.

**float64:distribution:negative_binomial:chf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the negative_binomial distribution:

**float64:distribution:negative_binomial:variance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the negative_binomial distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:negative_binomial:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the negative_binomial distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:negative_binomial:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the negative_binomial distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:negative_binomial:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for negative_binomial distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:negative_binomial:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the negative_binomial distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float64:distribution:negative_binomial:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the negative_binomial distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:laplace:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the laplace distribution.

**float64:distribution:laplace:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the laplace distribution.

**float64:distribution:laplace:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the laplace distribution.

**`float64:distribution:laplace:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the laplace distribution.

**`float64:distribution:laplace:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thelaplace distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:laplace:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the laplace distribution:

**`float64:distribution:laplace:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the laplace distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:laplace:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the laplace distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:laplace:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the laplace distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:laplace:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for laplace distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:laplace:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the laplace distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:laplace:kurtosisExcess`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the laplace distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:uniform:cdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the uniform distribution.

**`float64:distribution:uniform:pdf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the uniform distribution.

**`float64:distribution:uniform:cdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the uniform distribution.

**`float64:distribution:uniform:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the uniform distribution.

**`float64:distribution:uniform:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for theuniform distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:uniform:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the uniform distribution:

**`float64:distribution:uniform:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the uniform distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:uniform:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the uniform distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:uniform:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the uniform distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:uniform:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for uniform distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:uniform:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the uniform distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float64:distribution:uniform:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the uniform distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:fisher_f:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the fisher_f distribution.

**float64:distribution:fisher_f:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the fisher_f distribution.

**float64:distribution:fisher_f:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the fisher_f distribution.

**`float64:distribution:fisher_f:pdf:compl`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the fisher_f distribution.

**`float64:distribution:fisher_f:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thefisher_f distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:fisher_f:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the fisher_f distribution:

**`float64:distribution:fisher_f:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the fisher_f distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:fisher_f:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the fisher_f distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:fisher_f:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the fisher_f distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:fisher_f:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for fisher_f distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:fisher_f:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the fisher_f distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float64:distribution:fisher_f:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the fisher_f distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:weibull:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the weibull distribution.

**float64:distribution:weibull:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the weibull distribution.

**float64:distribution:weibull:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the weibull distribution.

**float64:distribution:weibull:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the weibull distribution.

**float64:distribution:weibull:quantile**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for theweibull distribution.
The last argument must be 0 < x <= 1.

**float64:distribution:weibull:chf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the weibull distribution:

**float64:distribution:weibull:variance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the weibull distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:weibull:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the weibull distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:weibull:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the weibull distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:weibull:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for weibull distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:weibull:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the weibull distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float64:distribution:weibull:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the weibull distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float64:distribution:pareto:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the pareto distribution.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x; \alpha, \beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the pareto distribution.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x; \alpha, \beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the pareto distribution.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$.
For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the pareto distribution.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$.
For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:quantile**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thepareto distribution.
The last argument must be 0 < x <= 1.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$.
For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:chf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the pareto distribution:
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$.
For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:variance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the pareto distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^\alpha}{x^{\alpha+1}}$.
For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the pareto distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the pareto distribution. The median function no parameters other than ones required to construct the distribution (see above).
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for pareto distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the pareto distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:pareto:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the pareto distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The Pareto distribution is a continuous distribution with the probability density function (pdf) $f(x;\alpha,\beta) = \frac{\alpha\beta^{\alpha}}{x^{\alpha+1}}$. For shape parameter $\alpha > 0$, and scale parameter $\beta > 0$ if $x < \beta$, then the pdf is zero. The Pareto distribution often describes the larger compared to the smaller. A classic example is that 80% of the wealth is owned by 20% of the population.

**float64:distribution:logistic:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the Distribution distribution.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the Distribution distribution.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the Distribution distribution.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the Distribution distribution.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:quantile**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for theDistribution distribution.
The last argument must be 0 < x <= 1.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:chf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the Distribution distribution:
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:variance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the Distribution distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the Distribution distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the Distribution distribution. The median function no parameters other than ones required to construct the distribution (see above).
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for Distribution distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the Distribution distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:logistic:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the Distribution distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The logistic distribution is a continous probability distribution. It has two parameters - location and scale. The cumulative distribution function of the logistic distribution appears in logistic regression and feedforward neural networks.Among other applications, United State Chess Federation and FIDE use it to calculate chess ratings.

**float64:distribution:lognormal:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: f$(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln x-m)^2}{2s^2}}$.

**float64:distribution:lognormal:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: f$(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln x-m)^2}{2s^2}}$.

**float64:distribution:lognormal:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:quantile**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for thelognormal distribution.
The last argument must be 0 < x <= 1.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:chf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the lognormal distribution:
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:variance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]

The variance for the lognormal distribution.

The variance function no parameters other than ones required to construct the distribution (see above).

The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.

For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:mean**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]

The mean for the lognormal distribution. The mean function no parameters other than ones required to construct the distribution (see above).

The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.

For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:median**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]

The median for the lognormal distribution. The median function no parameters other than ones required to construct the distribution (see above).

The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.

For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:standardDeviation**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]

The standard deviation for lognormal distribution.

The standard deviation function no parameters other than ones required to construct the distribution (see above).

The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.

For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float64:distribution:lognormal:kurtosis**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the lognormal distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln x-m)^2}{2s^2}}$.

**float64:distribution:lognormal:kurtosisExcess**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the lognormal distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: f$(x)$ = $\frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln x-m)^2}{2s^2}}$.

**float64:distribution:extreme_value:cdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the cumulative distribution function (cdf) for the extreme_value distribution.

**float64:distribution:extreme_value:pdf**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The probability density function function for the extreme_value distribution.

**float64:distribution:extreme_value:cdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Is the the complement $1-\text{cdf}(x)$ of the cumulative distribution function (cdf) for the extreme_value distribution.

**float64:distribution:extreme_value:pdf:compl**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The complement of the probability density function function for the extreme_value distribution.

**`float64:distribution:extreme_value:quantile`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The quantile function for theextreme_value distribution.
The last argument must be 0 < x <= 1.

**`float64:distribution:extreme_value:chf`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The chf function for the the extreme_value distribution:

**`float64:distribution:extreme_value:variance`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The variance for the extreme_value distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:extreme_value:mean`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The mean for the extreme_value distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:extreme_value:median`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The median for the extreme_value distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:extreme_value:standardDeviation`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The standard deviation for extreme_value distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:extreme_value:kurtosis`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis for the extreme_value distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float64:distribution:extreme_value:kurtosisExcess`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
The kurtosis excess for the extreme_value distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float64:distribution:students_t:findDegreesOfFreedom`**

*Keyspace* : float[64],float[64],float[64],float[64] *Valuespace* : float[64]
For the Student's T distribution find the number of degrees of freedom, given the four arguments: difference from the mean, alpha, beta and standard deviation.

**`float64:distribution:beta:estimate:fromMeanAndVariance:beta`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Estimate the beta parameter from mean, variance for the beta distribution.

**`float64:distribution:beta:estimate:fromMeanAndVariance:alpha`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Estimate the alpha parameter from mean, variance for the beta distribution.

**`float64:distribution:beta:estimate:fromProbability:beta`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Estimate the beta parameter from alpha parameter and probability for the beta distribution.

**`float64:distribution:beta:estimate:fromProbability:alpha`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
Estimate the alpha parameter from beta parameter and probability for the beta distribution.

**`float64:distribution:binomial:estimate:findMaximumNumberOfTrials`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:binomial:estimate:findMinimumNumberOfTrials`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:binomial:estimate:findLowerBoundOnP`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:binomial:estimate:findUpperBoundOnP`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:negative_binomial:estimate:findMaximumNumberOfTrials`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:negative_binomial:estimate:findMinimumNumberOfTrials`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:negative_binomial:estimate:findLowerBoundOnP`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:negative_binomial:estimate:findUpperBoundOnP`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:chi_squared:estimate:findDegreesOfFreedom`**

*Keyspace* : float[64],float[64],float[64],float[64] *Valuespace* : float[64]

**`float64:distribution:chi_squared:estimate:findDegreesOfFreedomHint`**

*Keyspace* : float[64],float[64],float[64],float[64],int[32] *Valuespace* : float[64]

**`float64:distribution:normal:findLocation`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The find_scale function from boost::math for the normal distribution.

**`float64:distribution:lognormal:findLocation`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The find_scale function from boost::math for the lognormal distribution.

**`float64:distribution:extreme_value:findLocation`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The find_scale function from boost::math for the extreme_value distribution.

**`float64:distribution:cauchy:findLocation`**

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
The find_scale function from boost::math for the cauchy distribution.

**`string:format:width:blox_float64`**

*Keyspace* : float[64],int[32] *Valuespace* : string
Takes a number of type blox_float64 and returns a string representation with a fixed width.

**`string:format:decimals:blox_float64`**

*Keyspace* : float[64],int[32],int[32] *Valuespace* : string
Takes a number of type blox_float64 and returns a string with overall length m and number of decimals n .

**`string:format:pad`**

*Keyspace* : string,int[32] *Valuespace* : string
Takes a string and a number and returns a the string padded to the number of characters. If the string is too large, it will be truncated; if it is too small it will be padded with spaces until the appropriate length is reached.

**`string:format:pad:left`**

*Keyspace* : string,int[32] *Valuespace* : string
Takes a string blox_string and a number and returns a string with the number length. If the string is too large it will truncated, and if it is too small it will be filled out with spaces so that it is left justified.

**`string:format:pad:right`**

*Keyspace* : string,int[32] *Valuespace* : string
Takes a string blox_string and a number and returns a string with the number length. If the string is too large it will truncated, and if it is too small it will be filled out with spaces so that it is right justified.

**`string:format:pad:internal`**

*Keyspace* : string,int[32] *Valuespace* : string
Takes a string blox_string and a number and returns a string with the number length. If the string is too large it will truncated, and if it is too small it will be filled out with spaces.

**`float64:nextAfter`**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns the next representable blox_float64 after the first argument in the direction of the second argument.

**`float64:floatAdvance`**

*Keyspace* : float[64],int[32] *Valuespace* : float[64]
Takes the first argument x of type blox_float64 and the second argument n of type blox_int32 and gives back nth representable blox_float64 after x.

**float64:floatDistance**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns numer of representable blox_float64 between the first and second argument. The result is also of the type blox_float64

**float64:min**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Takes x and y of type blox_float64 and returns the smaller one.

**float32:min**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Takes x and y of type blox_float32 and returns the smaller one.

**int32:min**

*Keyspace* : int[32],int[32] *Valuespace* : int[32]
Takes x and y of type blox_int32 and returns the smaller one.

**uint32:min**

*Keyspace* : uint[32],uint[32] *Valuespace* : uint[32]
Takes x and y of type blox_uint32 and returns the smaller one.

**int64:min**

*Keyspace* : int[64],int[64] *Valuespace* : int[64]
Takes x and y of type blox_int64 and returns the smaller one.

**uint64:min**

*Keyspace* : uint[64],uint[64] *Valuespace* : uint[64]
Takes x and y of type blox_uint64 and returns the smaller one.

**decimal64:min**

*Keyspace* : decimal[64],decimal[64] *Valuespace* : decimal[64]
Takes x and y of type blox_decimal64 and returns the smaller one.

**decimal128:min**

*Keyspace* : decimal[128],decimal[128] *Valuespace* : decimal[128]
Takes x and y of type blox_decimal128 and returns the smaller one.

**float64:max**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Takes x and y of type blox_float64 and returns the larger one.

**float32:max**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Takes x and y of type blox_float32 and returns the larger one.

**int32:max**

*Keyspace* : int[32],int[32] *Valuespace* : int[32]
Takes x and y of type blox_int32 and returns the larger one.

**uint32:max**

*Keyspace* : uint[32],uint[32] *Valuespace* : uint[32]
Takes x and y of type blox_uint32 and returns the larger one.

**int64:max**

*Keyspace* : int[64],int[64] *Valuespace* : int[64]
Takes x and y of type blox_int64 and returns the larger one.

**uint64:max**

*Keyspace* : uint[64],uint[64] *Valuespace* : uint[64]
Takes x and y of type blox_uint64 and returns the larger one.

**decimal64:max**

*Keyspace* : decimal[64],decimal[64] *Valuespace* : decimal[64]
Takes x and y of type blox_decimal64 and returns the larger one.

**decimal128:max**

*Keyspace* : decimal[128],decimal[128] *Valuespace* : decimal[128]
Takes x and y of type blox_decimal128 and returns the larger one.

**float64:tgamma**

*Keyspace* : float[64] *Valuespace* : float[64]
Returns the "true gamma" (hence name tgamma) of value z.

**float64:lgamma**

*Keyspace* : float[64] *Valuespace* : float[64]
Returns $\ln|\Gamma(z)|$ .

**float64:tgamma1pm1**

*Keyspace* : float[64] *Valuespace* : float[64]
Returns tgamma(dz + 1) - 1. Internally the implementation does not make use of the addition and subtraction implied by the definition, leading to accurate results even for very small dz. However, the implementation is capped to either 35 digit accuracy, or to the precision of the Lanczos approximation associated with type T, whichever is more accurate.

**float64:digamma**

*Keyspace* : float[64] *Valuespace* : float[64]
Returns the digamma or psi function of x. Digamma is defined as the logarithic derivative of the gamma function.

**float64:tgamma:ratio**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns the ratio of gamma function values: $\Gamma(a)/\Gamma(b)$.

**float64:tgamma:delta:ratio**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns the ratio of gamma function values: $\Gamma(a)/\Gamma(a+\delta)$ .

**float64:gamma:p**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns the normalised lower incomplete gamma `P(a,z)`, function of a and z. The function is normalized, i.e., the result is between 0 and 1. Requires `a > 0` and `z >= 0`.

**float64:gamma:q**

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns the normalised upper incomplete gamma function of a and z, Q(a,z). The function is normalized, i.e., the result is between 0 and 1. Requires a > 0 and z >= 0.

`float64:gamma:p:inv`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
No description.

`float64:gamma:p:inv:a`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
No description.

`float64:gamma:q:inv`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
Returns a value a such that: q = gamma:q[a, x].

`float64:gamma:q:inv:a`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
No description.

`float64:gamma:p:derivative`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
No description.

`float64:binomial:coefficient`

*Keyspace* : uint[32],uint[32] *Valuespace* : float[64]
No description.

`float64:beta`

*Keyspace* : float[64],float[64] *Valuespace* : float[64]
No description.

`float64:beta:incomplete:a`

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
No description.

`float64:beta:incomplete:c`

*Keyspace* : float[64],float[64],float[64] *Valuespace* : float[64]
No description.

**float64:erf**

*Keyspace* : float[64] *Valuespace* : float[64]
No description.

**float64:erf:inverse**

*Keyspace* : float[64] *Valuespace* : float[64]
No description.

**float64:financial:USD:string**

*Keyspace* : float[64] *Valuespace* : string
Takes floating point number and returns the string inUnited States Dollar financial format. Rounding is always to2digits.

**float64:financial:EUR:string**

*Keyspace* : float[64] *Valuespace* : string
Takes floating point number and returns the string inEuro financial format. Rounding is always to2digits.

**float32:distribution:students_t:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:pdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:cdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:pdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the students_t distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:quantile**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for thestudents_t distribution.
The last argument must be 0 < x <= 1.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the students_t distribution:
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:variance**

*Keyspace* : float[32] *Valuespace* : float[32]
The variance for the students_t distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:mean**

*Keyspace* : float[32] *Valuespace* : float[32]
The mean for the students_t distribution. The mean function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:median**

*Keyspace* : float[32] *Valuespace* : float[32]
The median for the students_t distribution. The median function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:standardDeviation**

*Keyspace* : float[32] *Valuespace* : float[32]
The standard deviation for students_t distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:kurtosis**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the students_t distribution. The kurtosis function no parameters other than ones required to construct the distribution.
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:students_t:kurtosisExcess**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the students_t distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
Student's T-distribution. The Student's t-distribution takes asingle parameter (v): the number of degrees of freedom of the sample. When the degrees of freedom is one then this distribution is the same asthe Cauchy-distribution. As the number of degrees of freedom tends towards infinity, then this distribution approaches the normal-distribution.

**float32:distribution:chi_squared:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

`float32:distribution:chi_squared:pdf`

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

`float32:distribution:chi_squared:cdf:compl`

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

`float32:distribution:chi_squared:pdf:compl`

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the chi_squared distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

`float32:distribution:chi_squared:quantile`

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for thechi_squared distribution.
The last argument must be 0 < x <= 1.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the chi_squared distribution:
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:variance**

*Keyspace* : float[32] *Valuespace* : float[32]
The variance for the chi_squared distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:mean**

*Keyspace* : float[32] *Valuespace* : float[32]
The mean for the chi_squared distribution. The mean function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:median**

*Keyspace* : float[32] *Valuespace* : float[32]
The median for the chi_squared distribution. The median function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:standardDeviation**

*Keyspace* : float[32] *Valuespace* : float[32]
The standard deviation for chi_squared distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:kurtosis**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the chi_squared distribution. The kurtosis function no parameters other than ones required to construct the distribution.
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:chi_squared:kurtosisExcess**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the chi_squared distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
A Chi-Squared distribution with $\nu$ degrees of freedom. The Chi-Squared distribution is one of the most widely used distributions in statistical tests. If $\chi$ are $\nu$ independent, normally distributed random variables with means $\mu$ and variances $\sigma$, then the random variable is distributed according to the Chi-Squared distribution.
The Chi-Squared distribution is a special case of the $\gamma$ distribution and has a single parameter $\nu$ that specifies the number of degrees of freedom.

**float32:distribution:exponential:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:pdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:cdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:pdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the exponential distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:quantile**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for theexponential distribution.
The last argument must be 0 < x <= 1.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the exponential distribution:
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:variance**

*Keyspace* : float[32] *Valuespace* : float[32]
The variance for the exponential distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:mean**

*Keyspace* : float[32] *Valuespace* : float[32]
The mean for the exponential distribution. The mean function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:median**

*Keyspace* : float[32] *Valuespace* : float[32]
The median for the exponential distribution. The median function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:standardDeviation**

*Keyspace* : float[32] *Valuespace* : float[32]
The standard deviation for exponential distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:kurtosis**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the exponential distribution. The kurtosis function no parameters other than ones required to construct the distribution.
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:exponential:kurtosisExcess**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the exponential distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
It is often used to model the time between independent events that happenat a constant average rate. Exponential distribution has one parameter, lambda.Lambda is defined as the reciprocal of the scale parameter.

**float32:distribution:rayleigh:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:pdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:cdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:pdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the rayleigh distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:quantile**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for therayleigh distribution.
The last argument must be 0 < x <= 1.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the rayleigh distribution:
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float32:distribution:rayleigh:variance`**

*Keyspace* : float[32] *Valuespace* : float[32]

The variance for the rayleigh distribution.

The variance function no parameters other than ones required to construct the distribution (see above).

The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float32:distribution:rayleigh:mean`**

*Keyspace* : float[32] *Valuespace* : float[32]

The mean for the rayleigh distribution. The mean function no parameters other than ones required to construct the distribution (see above).

The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float32:distribution:rayleigh:median`**

*Keyspace* : float[32] *Valuespace* : float[32]

The median for the rayleigh distribution. The median function no parameters other than ones required to construct the distribution (see above).

The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**`float32:distribution:rayleigh:standardDeviation`**

*Keyspace* : float[32] *Valuespace* : float[32]

The standard deviation for rayleigh distribution.

The standard deviation function no parameters other than ones required to construct the distribution (see above).

The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\text{exp}(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:kurtosis**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the rayleigh distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:rayleigh:kurtosisExcess**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the rayleigh distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The Rayleigh distribution is a continuous distribution with the probability density function: $f(x;\sigma) = \frac{x*\exp(-x^2/2\sigma^2)}{\sigma^2}$. The Rayleigh distribution is often used where two orthogonal components have an absolute value, for example, wind velocity and direction may be combined to yield a wind speed, or real and imaginary components may have absolute values that are Rayleigh distributed.

**float32:distribution:poisson:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:pdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:cdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:pdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the poisson distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:quantile**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for thepoisson distribution.
The last argument must be 0 < x <= 1.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the poisson distribution:
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:variance**

*Keyspace* : float[32] *Valuespace* : float[32]
The variance for the poisson distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:mean**

*Keyspace* : float[32] *Valuespace* : float[32]
The mean for the poisson distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:median**

*Keyspace* : float[32] *Valuespace* : float[32]
The median for the poisson distribution. The median function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:standardDeviation**

*Keyspace* : float[32] *Valuespace* : float[32]
The standard deviation for poisson distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:kurtosis**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the poisson distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:poisson:kurtosisExcess**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the poisson distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The Poisson distribution is a well-known statistical discrete distribution. It expresses the probability of a number of events (or failures, arrivals, occurrences ...) occurring in a fixed period of time, provided these events occur with a known mean rate $\lambda$ (events/time), and are independent of the time since the last event.
It has the Probability Mass Function f$(k;\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}$ for $k$ events, with an exepcted number of events $\lambda$.

**float32:distribution:bernoulli:cdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the bernoulli distribution.

**float32:distribution:bernoulli:pdf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The probability density function function for the bernoulli distribution.

**float32:distribution:bernoulli:cdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the bernoulli distribution.

**float32:distribution:bernoulli:pdf:compl**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the bernoulli distribution.

**float32:distribution:bernoulli:quantile**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The quantile function for thebernoulli distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:bernoulli:chf**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The chf function for the the bernoulli distribution:

**float32:distribution:bernoulli:variance**

*Keyspace* : float[32] *Valuespace* : float[32]
The variance for the bernoulli distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:bernoulli:mean**

*Keyspace* : float[32] *Valuespace* : float[32]
The mean for the bernoulli distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:bernoulli:median**

*Keyspace* : float[32] *Valuespace* : float[32]
The median for the bernoulli distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:bernoulli:standardDeviation`**

*Keyspace* : float[32] *Valuespace* : float[32]
The standard deviation for bernoulli distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:bernoulli:kurtosis`**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis for the bernoulli distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float32:distribution:bernoulli:kurtosisExcess`**

*Keyspace* : float[32] *Valuespace* : float[32]
The kurtosis excess for the bernoulli distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:normal:cdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the normal distribution.

**`float32:distribution:normal:pdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the normal distribution.

**`float32:distribution:normal:cdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the normal distribution.

**`float32:distribution:normal:pdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the normal distribution.

**`float32:distribution:normal:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thenormal distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:normal:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the normal distribution:

**float32:distribution:normal:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the normal distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:normal:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the normal distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:normal:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the normal distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:normal:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for normal distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:normal:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the normal distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:normal:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the normal distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:beta:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the beta distribution.

**`float32:distribution:beta:pdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the beta distribution.

**`float32:distribution:beta:cdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the beta distribution.

**`float32:distribution:beta:pdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the beta distribution.

**`float32:distribution:beta:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thebeta distribution.
The last argument must be 0 < x <= 1.

**`float32:distribution:beta:chf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the beta distribution:

**`float32:distribution:beta:variance`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the beta distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:beta:mean`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the beta distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:beta:median`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the beta distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:beta:standardDeviation`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for beta distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:beta:kurtosis`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the beta distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float32:distribution:beta:kurtosisExcess`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the beta distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:gamma:cdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the gamma distribution.

**`float32:distribution:gamma:pdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the gamma distribution.

**`float32:distribution:gamma:cdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the gamma distribution.

**`float32:distribution:gamma:pdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the gamma distribution.

**`float32:distribution:gamma:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thegamma distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:gamma:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the gamma distribution:

**float32:distribution:gamma:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the gamma distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:gamma:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the gamma distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:gamma:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the gamma distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:gamma:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for gamma distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:gamma:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the gamma distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:gamma:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the gamma distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:binomial:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the binomial distribution.

**`float32:distribution:binomial:pdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the binomial distribution.

**`float32:distribution:binomial:cdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the binomial distribution.

**`float32:distribution:binomial:pdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the binomial distribution.

**`float32:distribution:binomial:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thebinomial distribution.
The last argument must be 0 < x <= 1.

**`float32:distribution:binomial:chf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the binomial distribution:

**`float32:distribution:binomial:variance`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the binomial distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:binomial:mean`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the binomial distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:binomial:median`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the binomial distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:binomial:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for binomial distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:binomial:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the binomial distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:binomial:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the binomial distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:negative_binomial:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the negative_binomial distribution.

**float32:distribution:negative_binomial:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the negative_binomial distribution.

**float32:distribution:negative_binomial:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the negative_binomial distribution.

**float32:distribution:negative_binomial:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the negative_binomial distribution.

**float32:distribution:negative_binomial:quantile**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thenegative_binomial distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:negative_binomial:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the negative_binomial distribution:

**float32:distribution:negative_binomial:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the negative_binomial distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:negative_binomial:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the negative_binomial distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:negative_binomial:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the negative_binomial distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:negative_binomial:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for negative_binomial distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:negative_binomial:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the negative_binomial distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:negative_binomial:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the negative_binomial distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:laplace:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the laplace distribution.

**float32:distribution:laplace:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the laplace distribution.

**float32:distribution:laplace:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the laplace distribution.

**float32:distribution:laplace:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the laplace distribution.

**float32:distribution:laplace:quantile**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thelaplace distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:laplace:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the laplace distribution:

**float32:distribution:laplace:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the laplace distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:laplace:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the laplace distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:laplace:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the laplace distribution. The median function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:laplace:standardDeviation`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for laplace distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:laplace:kurtosis`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the laplace distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**`float32:distribution:laplace:kurtosisExcess`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the laplace distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**`float32:distribution:uniform:cdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the uniform distribution.

**`float32:distribution:uniform:pdf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the uniform distribution.

**`float32:distribution:uniform:cdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the uniform distribution.

**`float32:distribution:uniform:pdf:compl`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the uniform distribution.

**`float32:distribution:uniform:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for theuniform distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:uniform:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the uniform distribution:

**float32:distribution:uniform:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the uniform distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:uniform:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the uniform distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:uniform:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the uniform distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:uniform:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for uniform distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:uniform:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the uniform distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:uniform:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the uniform distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:fisher_f:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the fisher_f distribution.

**float32:distribution:fisher_f:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the fisher_f distribution.

**float32:distribution:fisher_f:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the fisher_f distribution.

**float32:distribution:fisher_f:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the fisher_f distribution.

**float32:distribution:fisher_f:quantile**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thefisher_f distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:fisher_f:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the fisher_f distribution:

**float32:distribution:fisher_f:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the fisher_f distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:fisher_f:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the fisher_f distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:fisher_f:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the fisher_f distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:fisher_f:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for fisher_f distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:fisher_f:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the fisher_f distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:fisher_f:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the fisher_f distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:weibull:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the weibull distribution.

**float32:distribution:weibull:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the weibull distribution.

**float32:distribution:weibull:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \text{cdf}(x)$ of the cumulative distribution function (cdf) for the weibull distribution.

**float32:distribution:weibull:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the weibull distribution.

**float32:distribution:weibull:quantile**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for theweibull distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:weibull:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the weibull distribution:

**float32:distribution:weibull:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the weibull distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:weibull:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the weibull distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:weibull:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the weibull distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:weibull:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for weibull distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:weibull:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the weibull distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:weibull:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the weibull distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:lognormal:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}\frac{-(\ln_{x-m})^2}{2s^2}$.

**float32:distribution:lognormal:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}\frac{-(\ln_{x-m})^2}{2s^2}$.

**float32:distribution:lognormal:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1 - \mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}\frac{-(\ln_{x-m})^2}{2s^2}$.

**float32:distribution:lognormal:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the lognormal distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $\mathrm{f}(x) = \frac{1}{xs\sqrt{2*\pi}}\frac{-(\ln_{x-m})^2}{2s^2}$.

**`float32:distribution:lognormal:quantile`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for thelognormal distribution.
The last argument must be 0 < x <= 1.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**`float32:distribution:lognormal:chf`**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the lognormal distribution:
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**`float32:distribution:lognormal:variance`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the lognormal distribution.
The variance function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**`float32:distribution:lognormal:mean`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the lognormal distribution. The mean function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float32:distribution:lognormal:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the lognormal distribution. The median function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float32:distribution:lognormal:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for lognormal distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float32:distribution:lognormal:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the lognormal distribution. The kurtosis function no parameters other than ones required to construct the distribution.
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float32:distribution:lognormal:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the lognormal distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).
The lognormal distribution is the distribution that arises when the logarithm ofthe random variable is normally distributed. A lognormal distribution results when the variable is the product of a large number of independent, identically-distributed variables.
For location and scale parameters m and s it is defined by the probability density function: $f(x) = \frac{1}{xs\sqrt{2*\pi}}^{\frac{-(\ln_{x-m})^2}{2s^2}}$.

**float32:distribution:extreme_value:cdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the cumulative distribution function (cdf) for the extreme_value distribution.

**float32:distribution:extreme_value:pdf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The probability density function function for the extreme_value distribution.

**float32:distribution:extreme_value:cdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Is the the complement $1-\mathrm{cdf}(x)$ of the cumulative distribution function (cdf) for the extreme_value distribution.

**float32:distribution:extreme_value:pdf:compl**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The complement of the probability density function function for the extreme_value distribution.

**float32:distribution:extreme_value:quantile**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The quantile function for theextreme_value distribution.
The last argument must be 0 < x <= 1.

**float32:distribution:extreme_value:chf**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The chf function for the the extreme_value distribution:

**float32:distribution:extreme_value:variance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The variance for the extreme_value distribution.
The variance function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:extreme_value:mean**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The mean for the extreme_value distribution. The mean function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:extreme_value:median**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The median for the extreme_value distribution. The median function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:extreme_value:standardDeviation**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The standard deviation for extreme_value distribution.
The standard deviation function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:extreme_value:kurtosis**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis for the extreme_value distribution. The kurtosis function no parameters other than ones required to construct the distribution.

**float32:distribution:extreme_value:kurtosisExcess**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
The kurtosis excess for the extreme_value distribution. The kurtosis excess function no parameters other than ones required to construct the distribution (see above).

**float32:distribution:students_t:findDegreesOfFreedom**

*Keyspace* : float[32],float[32],float[32],float[32] *Valuespace* : float[32]
For the Student's T distribution find the number of degrees of freedom, given the four arguments: difference from the mean, alpha, beta and standard deviation.

**float32:distribution:beta:estimate:fromMeanAndVariance:beta**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Estimate the beta parameter from mean, variance for the beta distribution.

**float32:distribution:beta:estimate:fromMeanAndVariance:alpha**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Estimate the alpha parameter from mean, variance for the beta distribution.

**float32:distribution:beta:estimate:fromProbability:beta**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Estimate the beta parameter from alpha parameter and probability for the beta distribution.

**float32:distribution:beta:estimate:fromProbability:alpha**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
Estimate the alpha parameter from beta parameter and probability for the beta distribution.

**float32:distribution:binomial:estimate:findMaximumNumberOfTrials**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:binomial:estimate:findMinimumNumberOfTrials**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:binomial:estimate:findLowerBoundOnP**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:binomial:estimate:findUpperBoundOnP**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:negative_binomial:estimate:findMaximumNumberOfTrials**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:negative_binomial:estimate:findMinimumNumberOfTrials**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:negative_binomial:estimate:findLowerBoundOnP**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:negative_binomial:estimate:findUpperBoundOnP**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:chi_squared:estimate:findDegreesOfFreedom**

*Keyspace* : float[32],float[32],float[32],float[32] *Valuespace* : float[32]

**float32:distribution:chi_squared:estimate:findDegreesOfFreedomHint**

*Keyspace* : float[32],float[32],float[32],float[32],int[32] *Valuespace* : float[32]

**float32:distribution:normal:findLocation**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The find_scale function from boost::math for the normal distribution.

**float32:distribution:lognormal:findLocation**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The find_scale function from boost::math for the lognormal distribution.

**float32:distribution:extreme_value:findLocation**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The find_scale function from boost::math for the extreme_value distribution.

**float32:distribution:cauchy:findLocation**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
The find_scale function from boost::math for the cauchy distribution.

**float32:nextAfter**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns the next representable blox_float32 after the first argument in the direction of the second argument.

**float32:floatAdvance**

*Keyspace* : float[32],int[32] *Valuespace* : float[32]
Takes the first argument x of type blox_float32 and the second argument n of type blox_int32 and gives back nth representable blox_float32 after x.

**float32:floatDistance**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns numer of representable blox_float32 between the first and second argument. The result is also of the type blox_float32

**float32:tgamma**

*Keyspace* : float[32] *Valuespace* : float[32]
Returns the “true gamma” (hence name tgamma) of value z.

**`float32:lgamma`**

*Keyspace* : float[64] *Valuespace* : float[32]
Returns $\ln|\Gamma(z)|$ .

**`float32:tgamma1pm1`**

*Keyspace* : float[32] *Valuespace* : float[32]
Returns tgamma(dz + 1) - 1. Internally the implementation does not make use of the addition and subtraction implied by the definition, leading to accurate results even for very small dz. However, the implementation is capped to either 35 digit accuracy, or to the precision of the Lanczos approximation associated with type T, whichever is more accurate.

**`float32:digamma`**

*Keyspace* : float[32] *Valuespace* : float[32]
Returns the digamma or psi function of x. Digamma is defined as the logarithic derivative of the gamma function.

**`float32:tgamma:ratio`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns the ratio of gamma function values: $\Gamma(a)/\Gamma(b)$.

**`float32:tgamma:delta:ratio`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns the ratio of gamma function values: $\Gamma(a)/\Gamma(a+\delta)$ .

**`float32:gamma:p`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns the normalised lower incomplete gamma `P(a,z)`, function of a and z. The function is normalized, i.e., the result is between 0 and 1. Requires `a > 0` and `z >= 0`.

**`float32:gamma:q`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns the normalised upper incomplete gamma function of a and z, Q(a,z). The function is normalized, i.e., the result is between 0 and 1. Requires a > 0 and z >= 0.

**`float32:gamma:p:inv`**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
No description.

**float32:gamma:p:inv:a**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
No description.

**float32:gamma:q:inv**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
Returns a value a such that: q = gamma:q[a, x].

**float32:gamma:q:inv:a**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
No description.

**float32:gamma:p:derivative**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
No description.

**float32:binomial:coefficient**

*Keyspace* : uint[32],uint[32] *Valuespace* : float[32]
No description.

**float32:beta**

*Keyspace* : float[32],float[32] *Valuespace* : float[32]
No description.

**float32:beta:incomplete:a**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
No description.

**float32:beta:incomplete:c**

*Keyspace* : float[32],float[32],float[32] *Valuespace* : float[32]
No description.

**float32:erf**

*Keyspace* : float[32] *Valuespace* : float[32]
No description.

**`float32:erf:inverse`**

*Keyspace* : float[32] *Valuespace* : float[32]
No description.

**`float32:financial:USD:string`**

*Keyspace* : float[32] *Valuespace* : string
Takes floating point number and returns the string inUnited States Dollar financial format. Rounding is always to2digits.

**`string:stem:spanish`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *spanish* language.

**`string:stem:russian`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *russian* language.

**`string:stem:romanian`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *romanian* language.

**`string:stem:portuguese`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *portuguese* language.

**`string:stem:porter`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *porter* language.

**`string:stem:norwegian`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *norwegian* language.

**`string:stem:italian`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *italian* language.

**`string:stem:hungarian`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *hungarian* language.

**`string:stem:german`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *german* language.

**`string:stem:french`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *french* language.

**`string:stem:finnish`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *finnish* language.

**`string:stem:english`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *english* language.

**`string:stem:dutch`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *dutch* language.

**`string:stem:danish`**

*Keyspace* : string *Valuespace* : string
This function takes a string (in UTF-8 format) and produces the word stem in *danish* language.

CHAPTER

SIX

# GLOSSARY

**constraint** A constraint is a restriction on the values that the variables may take in an optimization problem solution.

# BIBLIOGRAPHY

[BoostMath] .‘The Boost Math library‘. http://www.boost.org/doc/libs/1_44_0/libs/math/doc/html/index.html

# INDEX

## S



## T



## U



## W