PyStats

Release 0.1

Marc Izquierdo

Oct 25, 2021

GUIDE

PyStats is a Python 3 library of statistical analysis and distribution functions with simple **R**-like syntax, scalar/list input/output with OpenMP parallelization.

Author: Marc Izquierdo

License: MIT License

CHAPTER

ONE

INSTALLATION

You can use pip to install this library:

```
pip install pystats
```

Alternatively, you can also clone this repository and install the plugin manually using *pip*:

```
$ git clone git@github.com:marcizhu/PyStats.git
$ pip3 install ./PyStats
```

After that, you can just `import pystats` (or do `from pystats import *` if you don't want to type `pystats.` before all functions) and you're ready to go.

CHAPTER

TWO

CONTENTS

2.1 Syntax

Functions are called using an **R**-like syntax. Some general rules:

- Density functions: `pystats.d*`. For example, the Normal (Gaussian) density is called using

```
pystats.dnorm(<value>, <mean parameter>, <standard deviation>)
```

- Cumulative distribution functions: `pystats.p*`. For example, the Gamma CDF is called using

```
pystats.pgamma(<value>, <shape parameter>, <scale parameter>)
```

- Quantile functions: `pystats.q*`. For example, the Beta quantile is called using

```
pystats.qbeta(<value>, <a parameter>, <b parameter>)
```

- random sampling: `pystats.r*`. For example, to generate a single draw from the Logistic distribution:

```
pystats.rlogis(<location parameter>, <scale parameter>)
```

The library also supports lists as input/output:

- The pdf, cdf, and quantile functions can take list arguments. For example,

```
norm_pdf_vals = pystats.dnorm([x / 10 for x in range(-10, 10, 1)], 1.0, 2.0)
```

- The randomization functions (`r*`) can output random lists of arbitrary size. For example, the following code will generate a 100-item list of iid draws from a Gamma(3,2) distribution:

```
gamma_rvs = pystats.rgamma(100, 3.0, 2.0)
```

Additionally, most parameters have defaults to most common values and named parameters are also supported. For example, to generate a single draw from a Normal(0, 2) the following can be used:

```
norm_draw = pystats.rnorm(sd=2.0)
```

2.2 Examples

```
# Evaluate the normal PDF at x = 1, mu = 0, sigma = 1
dval_1 = pystats.dnorm(1.0, 0.0, 1.0)

# Evaluate the normal PDF at x = 1, mu = 0, sigma = 1, and return the log value
dval_2 = pystats.dnorm(1.0, 0.0, 1.0, True)

# Same as above, but using default values and named parameters
dval_3 = pystats.dnorm(1.0, log=True)

# Evaluate the normal CDF at x = 1, mu = 0, sigma = 1
pval = pystats.pnorm(1.0, 0.0, 1.0)

# Evaluate the Laplacian quantile at q = 0.1, mu = 0, sigma = 1
qval = pystats.qlaplace(0.1, 0.0, 1.0)

# Draw from a t-distribution with dof = 30
rval = pystats.rt(dof=30)

# List output
beta_rvs = pystats.rbeta(100, 3.0, 2.0)

# List input
beta_cdf_vals = pystats.pbeta(beta_rvs, 3.0, 2.0)
```

2.3 Distributions

2.3.1 Bernoulli Distribution

Table of contents

Density Function

The density function of the Bernoulli distribution:

$$f(x;p) = p^x (1-p)^{1-x} \times \mathbf{1}[x \in \{0,1\}]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dbern`(*x: int*, *prob: float*, *log: bool = False*) → float

Density function of the Bernoulli distribution.

Example

```
>>> pystats.dbern(1, 0.6)
0.6
```

Parameters

- **x** (*int*) – An integral-valued input, equal to 0 or 1.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dbern`(*x: List[int]*, *prob: float*, *log: bool = False*) → float

Density function of the Bernoulli distribution.

Example

```
>>> pystats.dbern([0, 1], 0.6)
[0.4, 0.6]
```

Parameters

- **x** (*List[int]*) – A standard list input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Bernoulli distribution:

$$F(x;p) = \sum_{z \leq x} f(z;p) = \begin{cases} 0 & \text{if } x < 0 \\ 1-p & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x \geq 1 \end{cases}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pbern`(*p: int*, *prob: float*, *log: bool = False*) → float
Distribution function of the Bernoulli distribution.

Example

```
>>> pystats.pbern(1, 0.6)
1.0
```

Parameters

- **p** (*int*) – A value equal to 0 or 1.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.pbern`(*p: List[int]*, *prob: float*, *log: bool = False*) → float
Distribution function of the Bernoulli distribution.

Example

```
>>> pystats.pbern([0, 1], 0.6)
[0.4, 1.0]
```

Parameters

- **p** (*List[int]*) – A standard list input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Bernoulli distribution:

$$q(r; p) = \begin{cases} 0 & \text{if } r \leq 1 - p \\ 1 & \text{else} \end{cases}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qbern`(*q: float*, *prob: float*) → float
Quantile function of the Bernoulli distribution.

Example

```
>>> pystats.qbern(0.5, 0.4)
0.0
```

Parameters

- **q** (*float*) – A real-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qbern`(*q: List[float]*, *prob: float*) → float
Quantile function of the Bernoulli distribution.

Example

```
>>> pystats.pbern([0.3, 0.7], 0.6)
[0.0, 1.0]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Bernoulli distribution is achieved via the inverse probability integral transform.

Scalar Output

`pystats.rbern(prob: float) → float`

Random sampling function for the Bernoulli distribution.

Example

```
>>> pystats.rbern(0.4)
0.0
```

Parameters **prob** (*float*) – The probability parameter, a real-valued input.

Returns A pseudo-random draw from the Bernoulli distribution.

List Output

`pystats.rbern(n: int, prob: float) → float`

Random sampling function for the Bernoulli distribution.

Example

```
>>> pystats.rbern(10, 0.4)
[1.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 0.0, 0.0]
```

Parameters

- **n** (*int*) – The number of output values.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns A list of pseudo-random draws from the Bernoulli distribution.

dbern	Density function of the Bernoulli distribution
pbern	Distribution function of the Bernoulli distribution
qbern	Quantile function of the Bernoulli distribution
rbern	Sampling function of the Bernoulli distribution

2.3.2 Beta Distribution

Table of contents

Density Function

The density function of the Beta distribution:

$$f(x; a, b) = \frac{1}{\mathcal{B}(a, b)} x^{a-1} (1-x)^{b-1} \times \mathbf{1}[0 \leq x \leq 1]$$

where $\mathcal{B}(a, b)$ denotes the Beta function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`dbeta`**(*x: float*, *shape1: float*, *shape2: float*, *log: bool = False*) → float

Density function of the Beta distribution.

Example

```
>>> pystats.dbeta(0.5, 3.0, 2.0)
1.5
```

Parameters

- **x** (*float*) – A real-valued input.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.

- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dbeta**(*x: List[float]*, *shape1: float*, *shape2: float*, *log: bool = False*) → List[float]
Density function of the Beta distribution.

Example

```
>>> pystats.dbeta([0.3, 0.5, 0.9], 3.0, 2.0)
[0.756, 1.5, 0.972]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Beta distribution:

$$F(x; a, b) = \int_0^x f(z; a, b) dz = I_x(a, b)$$

where $I_x(a, b)$ denotes the regularized incomplete Beta function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**pbeta**(*p: float*, *shape1: float*, *shape2: float*, *log: bool = False*) → float
Distribution function of the Beta distribution.

Example

```
>>> pystats.pbeta(0.5, 3.0, 2.0)
0.3125
```

Parameters

- **p** (*float*) – A real-valued input.

- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.`**`pbeta`**(*p: List[float]*, *shape1: float*, *shape2: float*, *log: bool = False*) → List[float]
Distribution function of the Beta distribution.

Example

```
>>> pystats.pbeta([0.3, 0.5, 0.9], 3.0, 2.0)
[0.0837, 0.3125, 0.9477]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Beta distribution:

$$q(p; a, b) = \inf \{x : p \leq I_x(a, b)\}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`qbeta`**(*q: float*, *shape1: float*, *shape2: float*) → float
Quantile function of the Beta distribution.

Example

```
>>> pystats.qbeta(0.5, 3.0, 2.0)
0.61427243186
```

Parameters

- **q** (*float*) – A real-valued input.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.

Returns The quantile function evaluated at q.

List Input

`pystats.`**`qbeta`**(*q: List[float]*, *shape1: float*, *shape2: float*) → List[float]
Quantile function of the Beta distribution.

Example

```
>>> pystats.qbeta([0.3, 0.5, 0.9], 3.0, 2.0)
[0.4915952451274149, 0.6142724318676113, 0.8574406832899696]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.

Returns A list of quantiles values corresponding to the elements of q.

Random Sampling

Random sampling for the Beta distribution is achieved by simulating two independent gamma-distributed random variables, $X \sim G(a, 1), Y \sim G(a, 1)$, then returning:

$$Z = \frac{X}{X + Y} \sim B(a, b)$$

Scalar Output

`pystats.rbeta(shape1: float, shape2: float) → float`

Random sampling function for the Beta distribution.

Example

```
>>> pystats.rbeta(3.0, 2.0)
0.3102933967927699
```

Parameters

- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.

Returns A pseudo-random draw from the Beta distribution.

List Output

`pystats.rbeta(n: int, shape1: float, shape2: float) → List[float]`

Random sampling function for the Beta distribution.

Example

```
>>> pystats.rbeta(5, 3.0, 2.0)
[0.2098009340766354, 0.483502928842804, 0.5101449459492882, 0.46683636135288137, 0.
↪791214142230094]
```

Parameters

- **n** (*int*) – The number of output values.
- **shape1** (*float*) – A real-valued shape parameter.
- **shape2** (*float*) – A real-valued shape parameter.

Returns A list of pseudo-random draws from the Beta distribution.

dbeta	Density function of the Beta distribution
pbeta	Distribution function of the Beta distribution
qbeta	Quantile function of the Beta distribution
rbeta	Sampling function of the Beta distribution

2.3.3 Binomial Distribution

Table of contents

Density Function

The density function of the Binomial distribution:

$$f(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x} \times \mathbf{1}[x \in \{0, \ldots, n\}]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dbinom`(*x: float*, *n_trials: int*, *prob: float*, *log: bool = False*) → float

Density function of the Binomial distribution.

Example

```
>>> pystats.dbinom(2, 4, 0.4)
0.3456
```

Parameters

- **x** (*int*) – An integral-valued input, equal to 0 or 1.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dbinom**(*x: List[float]*, *n_trials: int*, *prob: float*, *log: bool = False*) → List[float]

Density function of the Binomial distribution.

Example

```
>>> pystats.dbinom([2, 3, 4], 5, 0.4)
[0.3456, 0.2304, 0.0768]
```

Parameters

- **x** (*List[int]*) – A standard list input.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Binomial distribution:

$$F(x; n, p) = \sum_{z \leq x} f(z; n, p)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**pbinom**(*p: float*, *n_trials: int*, *prob: float*, *log: bool = False*) → float

Distribution function of the Binomial distribution.

Example

```
>>> pystats.pbinom(2, 4, 0.4)
0.8208
```

Parameters

- **p** (*float*) – A value equal to 0 or 1.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

pystats.**pbinom**(*p: List[float]*, *n_trials: int*, *prob: float*, *log: bool = False*) → List[float]
Distribution function of the Binomial distribution.

Example

```
>>> pystats.pbinom([2, 3, 4], 5, 0.4)
[0.68256, 0.91296, 0.98976]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Binomial distribution:

$$q(r; n, p) = \inf \{x : r \leq F(x; n, p)\}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qbinom**(*q: float*, *n_trials: int*, *prob: float*) → float
Quantile function of the Binomial distribution.

Example

```
>>> pystats.qbinom(0.5, 4, 0.4)
2
```

Parameters

- **q** (*float*) – A real-valued input.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qbinom**(*q: List[float]*, *n_trials: int*, *prob: float*) → List[float]

Quantile function of the Binomial distribution.

Example

```
>>> pystats.qbinom([0.2, 0.4, 0.8], 5, 0.4)
[1, 2, 3]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Binomial distribution is achieved by summing the results of simulating *n* Bernoulli-distributed random variables.

Scalar Output

pystats.**rbinom**(*n_trials: int*, *prob: float*) → float

Random sampling function for the Binomial distribution.

Example

```
>>> pystats.rbinom(4, 0.4)
2
```

Parameters

- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns A pseudo-random draw from the Binomial distribution.

List Output

pystats.**rbinom**(*n: int*, *n_trials: int*, *prob: float*) → List[float]

Random sampling function for the Binomial distribution.

Example

```
>>> pystats.rbinom(10, 4, 0.4)
[1, 4, 0, 2, 3, 2, 2, 2, 2, 1]
```

Parameters

- **n** (*int*) – The number of output values.
- **n_trials** (*int*) – The number of trials, a non-negative integral-valued input.
- **prob** (*float*) – The probability parameter, a real-valued input.

Returns A list of pseudo-random draws from the Binomial distribution.

dbinom	Density function of the Binomial distribution
pbinom	Distribution function of the Binomial distribution
qbinom	Quantile function of the Binomial distribution
rbinom	Sampling function of the Binomial distribution

2.3.4 Cauchy Distribution

Table of contents

Density Function

The density function of the Cauchy distribution:

$$f(x; \mu, \sigma) = \frac{1}{\pi\sigma \left[1 + \left(\frac{x-\mu}{\sigma}\right)^2\right]}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dcauchy`(*x: float*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → float

Density function of the Cauchy distribution.

Example

```
>>> pystats.dcauchy(2.5, 1.0, 3.0)
0.084883
```

Parameters

- **x** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dcauchy`(*x: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]

Density function of the Cauchy distribution.

Example

```
>>> pystats.dcauchy([0.0, 1.0, 2.0], 1.0, 2.0)
[0.12732395447351627, 0.15915494309189535, 0.12732395447351627]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Cauchy distribution:

$$F(x;\mu,\sigma) = \int_{-\infty}^{x} f(z;\mu,\sigma) dz = 0.5 + \frac{1}{\pi}\text{arctan}\left(\frac{x - \mu}{\sigma} \right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`pcauchy`**(*p: float*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → float

Distribution function of the Cauchy distribution.

Example

```
>>> pystats.pcauchy(2.5, 1.0, 3.0)
0.647584
```

Parameters

- **p** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.`**`pcauchy`**(*p: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Cauchy distribution.

Example

```
>>> pystats.pcauchy([0.0, 1.0, 2.0], 1.0, 2.0)
[0.35241638234956674, 0.5, 0.6475836176504333]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Cauchy distribution:

$$q(p; \mu, \sigma) = \mu + \gamma \text{tan}\left(\pi(p - 0.5)\right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qcauchy**(*q: float*, *mu: float = 0.0*, *sigma: float = 1.0*) → float
Quantile function of the Cauchy distribution.

Example

```
>>> pystats.qcauchy(0.5, 1, 3.0)
0.647584
```

Parameters

- **q** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qcauchy**(*q: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*) → List[float]
Quantile function of the Cauchy distribution.

Example

```
>>> pystats.qcauchy([0.1, 0.3, 0.7], 1.0, 2.0)
[-5.155367074350508, -0.45308505601072185, 2.453085056010721]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Cauchy distribution is achieved via the inverse probability integral transform.

Scalar Output

pystats.**rcauchy**(*mu: float = 0.0*, *sigma: float = 1.0*) → float

Random sampling function for the Cauchy distribution.

Example

```
>>> pystats.rcauchy(1.0, 2.0)
9.93054237677352
```

Parameters

- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A pseudo-random draw from the Cauchy distribution.

List Output

pystats.**rcauchy**(*n: int*, *mu: float = 0.0*, *sigma: float = 1.0*) → List[float]

Random sampling function for the Cauchy distribution.

Example

```
>>> pystats.rcauchy(1.0, 2.0)
[-2.383182638662492, 1.0766564460128407, -20.367599105297693, -0.9512379893292959, -
↪0.17185207327053853]
```

Parameters

- **n** (*int*) – The number of output values.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of pseudo-random draws from the Cauchy distribution.

dcauchy	Density function of the Cauchy distribution
pcauchy	Distribution function of the Cauchy distribution
qcauchy	Quantile function of the Cauchy distribution
rcauchy	Sampling function of the Cauchy distribution

2.3.5 Chi-squared Distribution

Table of contents

Density Function

The density function of the Chi-squared distribution:

$$f(x;k) = \frac{x^{k/2-1}\exp(-x/2)}{2^{k/2}\Gamma(k/2)} \times \mathbf{1}[x \geq 0]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dchisq`(*x: float*, *dof: float = 1.0*, *log: bool = False*) → float

Density function of the Chi-squared distribution.

Example

```
>>> pystats.dchisq(4.0, 5)
0.1439759107018347
```

Parameters

- **x** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dchisq(x: List[float], dof: float = 1.0, log: bool = False) → List[float]`

Density function of the Chi-squared distribution.

Example

```
>>> pystats.dchisq([1.8, 0.7, 4.2], 4)
[0.18295634688326964, 0.12332041570077489, 0.12857924966563097]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Chi-squared distribution:

$$F(x; k) = \int_0^x f(z; k) dz = \frac{\gamma(k/2, x/2)}{\Gamma(k/2)}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pchisq(p: float, dof: float = 1.0, log: bool = False) → float`

Distribution function of the Chi-squared distribution.

Example

```
>>> pystats.pchisq(4.0, 5)
0.45058404864721946
```

Parameters

- **p** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

pystats.**pchisq**(*p: List[float]*, *dof: float = 1.0*, *log: bool = False*) → List[float]
Distribution function of the Chi-squared distribution.

Example

```
>>> pystats.pchisq([1.8, 0.7, 4.2], 4)
[0.22751764649286174, 0.048671078879736845, 0.620385072415756]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Chi-squared distribution:

$$q(p; k) = \inf \{x : p \leq \gamma(k/2, x/2)/\Gamma(k/2)\}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qchisq**(*q: float*, *dof: float = 1.0*) → float
Quantile function of the Chi-squared distribution.

Example

```
>>> pystats.qchisq(0.5, 5)
4.351460191095529
```

Parameters

- **q** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qchisq**(*q: List[float]*, *dof: float = 1.0*) → List[float]
Quantile function of the Chi-squared distribution.

Example

```
>>> pystats.qchisq([1.8, 0.7, 4.2], 4)
[2.194698421406983, 3.356693980033322, 5.988616694004245]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Scalar Output

pystats.**rchisq**(*dof: float = 1.0*) → float
Random sampling function for the Chi-squared distribution.

Example

```
>>> pystats.rchisq(dof=5)
7.088454619471778
```

Parameters **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A pseudo-random draw from the Chi-squared distribution.

List Output

pystats.**rchisq**(*n: int*, *dof: float = 1.0*) → List[float]
Random sampling function for the Chi-squared distribution.

Example

```
>>> pystats.rchisq(5, 5)
[2.3284093401299866, 9.215161276152928, 6.904990781549569, 8.257146493760509, 4.
↪299710184814277]
```

Parameters

- **n** (*int*) – The number of output values.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A list of pseudo-random draws from the Chi-squared distribution.

dchisq	Density function of the Chi-squared distribution
pchisq	Distribution function of the Chi-squared distribution
qchisq	Quantile function of the Chi-squared distribution
rchisq	Sampling function of the Chi-squared distribution

2.3.6 Exponential Distribution

Table of contents

Density Function

The density function of the Exponential distribution:

$$f(x; \lambda) = \lambda \exp(-\lambda x) \times \mathbf{1}[x \geq 0]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**dexp**(*x: float*, *rate: float = 1.0*, *log: bool = False*) → float

Density function of the Exponential distribution.

Example

```
>>> pystats.dexp(1.0, 2.0)
0.2706705664732254
```

Parameters

- **x** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dexp**(*x: List[float]*, *rate: float = 1.0*, *log: bool = False*) → List[float]

Density function of the Exponential distribution.

Example

```
>>> pystats.dexp([1.8, 0.7, 4.2], 4.0)
[0.0029863432335067172, 0.24324025050087195, 2.022612539334209e-07]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Exponential distribution:

$$\int_0^x f(z; \lambda) dz = 1 - \exp(-\lambda x \times \mathbf{1}[x \geq 0])$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`pexp`**(*p: float*, *rate: float = 1.0*, *log: bool = False*) → float

Distribution function of the Exponential distribution.

Example

```
>>> pystats.pexp(1.0, 2.0)
0.8646647167633873
```

Parameters

- **p** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.`**`pexp`**(*p: List[float]*, *rate: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Exponential distribution.

Example

```
>>> pystats.pexp([1.8, 0.7, 4.2], 4.0)
[0.9992534141916233, 0.9391899373747821, 0.9999999494346865]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Exponential distribution:

$$q(p; \lambda) = -\ln(1-p)/\lambda$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qexp(q: float, rate: float = 1.0)` → float

Quantile function of the Exponential distribution.

Example

```
>>> pystats.qexp(0.5, 2.0)
0.3465735902799726
```

Parameters

- **q** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qexp(q: List[float], rate: float = 1.0)` → List[float]

Quantile function of the Exponential distribution.

Example

```
>>> pystats.qexp([0.3, 0.5, 0.8], 4.0)
[0.08916873598468311, 0.1732867951399863, 0.40235947810852524]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Cauchy distribution is achieved via the inverse probability integral transform.

Scalar Output

`pystats.rexp(rate: float = 1.0) → float`

Random sampling function for the Exponential distribution.

Example

```
>>> pystats.rexp(2.0)
0.8337215251612762
```

Parameters `rate` (*float*) – The rate parameter, a real-valued input.

Returns A pseudo-random draw from the Exponential distribution.

List Output

`pystats.rexp(n: int, rate: float = 1.0) → List[float]`

Random sampling function for the Exponential distribution.

Example

```
>>> pystats.rexp(3, 2.0)
[0.006095192297017023, 0.552560396122137, 0.8185248559121117]
```

Parameters

- `n` (*int*) – The number of output values.
- `rate` (*float*) – The rate parameter, a real-valued input.

Returns A list of pseudo-random draws from the Exponential distribution.

dexp	Density function of the Exponential distribution
pexp	Distribution function of the Exponential distribution
qexp	Quantile function of the Exponential distribution
rexp	Sampling function of the Exponential distribution

2.3.7 F-Distribution

Table of contents

Density Function

The density function of the F distribution:

$$f(x; d_1, d_2) = \frac{1}{\mathcal{B}\left(\frac{d_1}{2}, \frac{d_2}{2}\right)} \left(\frac{d_1}{d_2}\right)^{\frac{d_1}{2}} x^{d_1/2-1} \left(1 + \frac{d_1}{d_2} x\right)^{-\frac{d_1+d_2}{2}} \times \mathbf{1}[x \geq 0]$$

where $\mathcal{B}(a, b)$ denotes the Beta function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`df`**(*x: float*, *df1: float = 1.0*, *df2: float = 1.0*, *log: bool = False*) → float

Density function of the F-distribution distribution.

Example

```
>>> pystats.df(1.5, 10.0, 12.0)
0.3426270538333347
```

Parameters

- **x** (*float*) – A real-valued input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.

- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.df(x: List[float], df1: float = 1.0, df2: float = 1.0, log: bool = False) → List[float]`
Density function of the F-distribution distribution.

Example

```
>>> pystats.df([0.3, 0.5, 0.9], 10.0, 12.0)
[0.3523215359999982, 0.6861197229627093, 0.7047187344898975]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the F distribution:

$$F(x; d_1, d_2) = \int_0^x f(z; d_1, d_2) dz = I_{\frac{d_1 x}{d_2 + d_1 x}}(d_1/2, d_2/2)$$

where $I_x(a, b)$ denotes the regularized incomplete Beta function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pf(p: float, df1: float = 1.0, df2: float = 1.0, log: bool = False) → float`
Distribution function of the F-distribution distribution.

Example

```
>>> pystats.pf(1.5, 10.0, 12.0)
0.7501297253279772
```

Parameters

- **p** (*float*) – A real-valued input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

pystats.**pf**(*p: List[float]*, *df1: float = 1.0*, *df2: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the F-distribution distribution.

Example

```
>>> pystats.pf([0.3, 0.5, 0.9], 10.0, 12.0)
[0.03279349759999985, 0.14036282082102308, 0.43990811032084115]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the F distribution:

$$q(p; a, b) = \inf \left\{ x : p \leq I_{\frac{d_1 x}{d_2 + d_1 x}}(d_1/2, d_2/2) \right\}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qf**(*q: float*, *df1: float = 1.0*, *df2: float = 1.0*) → float
Quantile function of the F-distribution distribution.

Example

```
>>> pystats.qf(0.5, 10.0, 12.0)
0.9885595669294069
```

Parameters

- **q** (*float*) – A real-valued input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.

Returns The quantile function evaluated at q.

List Input

pystats.**qf**(*q: List[float]*, *df1: float = 1.0*, *df2: float = 1.0*) → List[float]
Quantile function of the F-distribution distribution.

Example

```
>>> pystats.qf([0.3, 0.5, 0.9], 10.0, 12.0)
[0.7125992144145168, 0.9885595669294069, 2.1877640788750874]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of q.

Random Sampling

Random sampling for the Beta distribution is achieved by simulating two independent χ^2-distributed random variables, $X \sim \chi^2(d_1), Y \sim \chi^2(d_2)$, then returning:

$$Z = \frac{d_1}{d_2} \frac{X}{Y}$$

Scalar Output

`pystats.rf(df1: float = 1.0, df2: float = 1.0) → float`

Random sampling function for the F-distribution distribution.

Example

```
>>> pystats.rf(10.0, 12.0)
2.646874442851056
```

Parameters

- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.

Returns A pseudo-random draw from the F-distribution distribution.

List Output

`pystats.rf(n: int, df1: float = 1.0, df2: float = 1.0) → List[float]`

Random sampling function for the F-distribution distribution.

Example

```
>>> pystats.rf(3, 10.0, 12.0)
[0.4932413364221738, 0.2827026830899671, 2.750316525821291]
```

Parameters

- **n** (*int*) – The number of output values.
- **df1** (*float*) – A degrees of freedom parameter, a real-valued input.
- **df2** (*float*) – A degrees of freedom parameter, a real-valued input.

Returns A list of pseudo-random draws from the F-distribution distribution.

df	Density function of the F-distribution
pf	Distribution function of the F-distribution
qf	Quantile function of the F-distribution
rf	Sampling function of the F-distribution

2.3.8 Gamma Distribution

Table of contents

Density Function

The density function of the Gamma distribution:

$$f(x; k, \theta) = \frac{x^{k-1} \exp(-x/\theta)}{\theta^k \Gamma(k)} \times \mathbf{1}[x \geq 0]$$

where $\Gamma(\cdot)$ denotes the Gamma function, k is the shape parameter, and θ is the scale parameter.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`dgamma`**(*x: float*, *shape: float = 1.0*, *scale: float = 1.0*, *log: bool = False*) → float

Density function of the Gamma distribution.

Example

```
>>> pystats.dgamma(2, 2, 3)
0.1140926931183538
```

Parameters

- **x** (*float*) – A real-valued input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.

- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dgamma`(*x: List[float]*, *shape: float = 1.0*, *scale: float = 1.0*, *log: bool = False*) → List[float]
Density function of the Gamma distribution.

Example

```
>>> pystats.dgamma([1.8, 0.7, 4.2], 2, 3)
[0.1097623272188053, 0.061591410715083, 0.11507858317274963]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Gamma distribution:

$$F(x; k, \theta) = \int_0^x f(z; k, \theta) dz = \frac{\gamma(k, x\theta)}{\Gamma(k)}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pgamma`(*p: float*, *shape: float = 1.0*, *scale: float = 1.0*, *log: bool = False*) → float
Distribution function of the Gamma distribution.

Example

```
>>> pystats.pgamma(2, 2, 3)
0.14430480161234657
```

Parameters

- **p** (*float*) – A real-valued input.

- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.`**`pgamma`**(*p: List[float]*, *shape: float = 1.0*, *scale: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Gamma distribution.

Example

```
>>> pystats.pgamma([1.8, 0.7, 4.2], 2, 3)
[0.12190138224955768, 0.023336201517969262, 0.4081672865401445]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Gamma distribution:

$$q(p; k, \theta) = \inf \left\{ x : p \leq \frac{\gamma(k, x\theta)}{\Gamma(k)} \right\}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`qgamma`**(*q: float*, *shape: float = 1.0*, *scale: float = 1.0*) → float

Quantile function of the Gamma distribution.

Example

```
>>> pystats.qgamma(0.15, 2, 3)
2.0497158392128205
```

Parameters

- **q** (*float*) – A real-valued input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qgamma**(*q: List[float]*, *shape: float = 1.0*, *scale: float = 1.0*) → List[float]
Quantile function of the Gamma distribution.

Example

```
>>> pystats.qgamma([0.1, 0.2, 0.7], 2, 3)
[1.5954348251688362, 2.473164927098954, 7.317649449840613]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Gamma distribution is achieved via the Ziggurat method of Marsaglia and Tsang (2000).

Scalar Output

pystats.**rgamma**(*shape: float = 1.0*, *scale: float = 1.0*) → float
Random sampling function for the Gamma distribution.

Example

```
>>> pystats.rgamma(2, 3)
2.5950379008163194
```

Parameters

- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.

Returns A pseudo-random draw from the Gamma distribution.

List Output

`pystats.rgamma`(*n: int*, *shape: float = 1.0*, *scale: float = 1.0*) → List[float]

Random sampling function for the Gamma distribution.

Example

```
>>> pystats.rgamma(3, 2, 3)
[8.584541442906463, 4.491138145011711, 4.904685252957054]
```

Parameters

- **n** (*int*) – The number of output values.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **scale** (*float*) – The scale parameter, a real-valued input.

Returns A list of pseudo-random draws from the Gamma distribution.

dgamma	Density function of the Gamma distribution
pgamma	Distribution function of the Gamma distribution
qgamma	Quantile function of the Gamma distribution
rgamma	Sampling function of the Gamma distribution

2.3.9 Inverse-Gamma Distribution

Table of contents

Density Function

The density function of the inverse-Gamma distribution:

$$f(x;\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{-\alpha-1} \exp\left(-\frac{\beta}{x}\right) \times \mathbf{1}[x \geq 0]$$

where $\Gamma(\cdot)$ denotes the Gamma function, α is the shape parameter, and β is the rate parameter.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`dinvgamma`**(*x: float*, *shape: float = 1.0*, *rate: float = 1.0*, *log: bool = False*) → float

Density function of the inverse-Gamma distribution.

Example

```
>>> pystats.dinvgamma(1.5, 2, 1)
0.15212359082447174
```

Parameters

- **x** (*float*) – A real-valued input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dinvgamma**(*x: List[float]*, *shape: float = 1.0*, *rate: float = 1.0*, *log: bool = False*) → List[float]
Density function of the inverse-Gamma distribution.

Example

```
>>> pystats.dinvgamma([1.8, 0.7, 4.2], 3, 2)
[0.12543552347504414, 0.9568116496062874, 0.007984650912112904]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the inverse-Gamma distribution:

$$F(x; \alpha, \beta) = \int_0^x f(z; \alpha, \beta) dz = 1 - \frac{\gamma(1/x, \beta/x)}{\Gamma(\alpha)}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**pinvgamma**(*p: float*, *shape: float = 1.0*, *rate: float = 1.0*, *log: bool = False*) → float
Distribution function of the inverse-Gamma distribution.

Example

```
>>> pystats.pinvgamma(1.5, 2, 1)
0.8556951983876534
```

Parameters

- **p** (*float*) – A real-valued input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.pinvgamma(p: List[float], shape: float = 1.0, rate: float = 1.0, log: bool = False)` → List[float]

Distribution function of the inverse-Gamma distribution.

Example

```
>>> pystats.pinvgamma([1.8, 0.7, 4.2], 3, 2)
[0.8981685222907052, 0.4559446713286356, 0.9873531870485975]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the inverse-Gamma distribution:

$$q(p; \alpha, \beta) = \inf \left\{ x : p \leq 1 - \frac{\gamma(1/x, \beta/x)}{\Gamma(\alpha)} \right\}$$

where $\Gamma(\cdot)$ denotes the gamma function and $\gamma(\cdot, \cdot)$ denotes the incomplete gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qinvgamma(q: float, shape: float = 1.0, rate: float = 1.0)` → float

Quantile function of the inverse-Gamma distribution.

Example

```
>>> pystats.qinvgamma(0.95, 2, 1)
2.8140357632821265
```

Parameters

- **q** (*float*) – A real-valued input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qinvgamma**(*q: List[float]*, *shape: float = 1.0*, *rate: float = 1.0*) → List[float]

Quantile function of the inverse-Gamma distribution.

Example

```
>>> pystats.qinvgamma([0.3, 0.5, 0.6], 3, 2)
[0.5531634821501723, 0.7479262863802247, 0.87524406574503711]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of q.

Random Sampling

Random sampling for the inverse-Gamma distribution is achieved by simulating $X \sim G(\alpha, 1/\beta)$, then returning

$$Z = \frac{1}{X} \sim \mathrm{IG}(\alpha, \beta)$$

Scalar Output

pystats.**rinvgamma**(*shape: float = 1.0*, *rate: float = 1.0*) → float

Random sampling function for the inverse-Gamma distribution.

Example

```
>>> pystats.rinvgamma(2, 1)
0.29563080448131196
```

Parameters

- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A pseudo-random draw from the inverse-Gamma distribution.

List Output

pystats.**rinvgamma**(*n: int*, *shape: float = 1.0*, *rate: float = 1.0*) → List[float]

Random sampling function for the inverse-Gamma distribution.

Example

```
>>> pystats.rinvgamma(3, 2, 1)
[0.31841204990923705, 0.47383794440642224, 0.4720582119984054]
```

Parameters

- **n** (*int*) – The number of output values.
- **shape** (*float*) – The shape parameter, a real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A list of pseudo-random draws from the inverse-Gamma distribution.

dinvgamma	Density function of the inverse Gamma distribution
pinvgamma	Distribution function of the inverse Gamma distribution
qinvgamma	Quantile function of the inverse Gamma distribution
rinvgamma	Sampling function of the inverse Gamma distribution

2.3.10 Laplace Distribution

Table of contents

Density Function

The density function of the Laplace distribution:

$$f(x; \mu, \sigma) = \frac{1}{2\sigma} \exp\left(- \frac{|x - \mu|}{\sigma} \right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**dlaplace**(*x: float*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → float
Density function of the Laplace distribution.

Example

```
>>> pystats.dlaplace(0.7, 1.0, 2.0)
0.21517699410626442
```

Parameters

- **x** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dlaplace**(*x: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]
Density function of the Laplace distribution.

Example

```
>>> pystats.dlaplace([0.0, 1.0, 2.0], 1.0, 2.0)
[0.15163266492815836, 0.25, 0.15163266492815836]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Laplace distribution:

$$F(x;\mu,\sigma) = \int_{-\infty}^{x} f(z;\mu,\sigma) dz = \frac{1}{2} + \frac{1}{2} \times \text{sign}(x-\mu) \times \left(1 - \exp\left(-\frac{|x-\mu|}{\sigma}\right)\right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.plaplace(p: float, mu: float = 0.0, sigma: float = 1.0, log: bool = False)` → float

Distribution function of the Laplace distribution.

Example

```
>>> pystats.plaplace(0.7, 1.0, 2.0)
0.4303539882125289
```

Parameters

- **p** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.plaplace(p: List[float], mu: float = 0.0, sigma: float = 1.0, log: bool = False)` → List[float]

Distribution function of the Laplace distribution.

Example

```
>>> pystats.plaplace([0.0, 1.0, 2.0], 1.0, 2.0)
[0.3032653298563167, 0.5, 0.6967346701436833]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Laplace distribution:

$$q(p; \mu, \sigma) = \mu - \sigma \times \text{sign}(p - 0.5) \times \ln(1 - 2|p - 0.5|)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qlaplace(q: float, mu: float = 0.0, sigma: float = 1.0)` → float

Quantile function of the Laplace distribution.

Example

```
>>> pystats.qlaplace(0.5, 1.0, 2.0)
1.0
```

Parameters

- **q** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns The quantile function evaluated at q.

List Input

`pystats.qlaplace(q: List[float], mu: float = 0.0, sigma: float = 1.0)` → List[float]

Quantile function of the Laplace distribution.

Example

```
>>> pystats.qlaplace([0.1, 0.6, 0.9], 1.0, 2.0)
[-2.218875824868202, 1.4462871026284194, 4.218875824868202]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of q.

Random Sampling

Random sampling for the Laplace distribution is achieved via the inverse probability integral transform.

Scalar Output

`pystats.`**`rlaplace`**`(`*mu: float = 0.0*, *sigma: float = 1.0*`)` → float

Random sampling function for the Laplace distribution.

Example

```
>>> pystats.rlaplace(1.0, 2.0)
-0.06931476521956581
```

Parameters

- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A pseudo-random draw from the Laplace distribution.

List Output

`pystats.`**`rlaplace`**`(`*n: int*, *mu: float = 0.0*, *sigma: float = 1.0*`)` → List[float]

Random sampling function for the Laplace distribution.

Example

```
>>> pystats.rlaplace(3, 1.0, 2.0)
[0.7785860236771809, 0.14724988410675632, 3.0446760367044163]
```

Parameters

- **n** (*int*) – The number of output values.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of pseudo-random draws from the Laplace distribution.

dlaplace	Density function of the Laplace distribution
plaplace	Distribution function of the Laplace distribution
qlaplace	Quantile function of the Laplace distribution
rlaplace	Sampling function of the Laplace distribution

2.3.11 Log-Normal Distribution

Table of contents

Density Function

The density function of the log-Normal distribution:

$$f(x; \mu, \sigma) = \frac{1}{x} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{(\ln x - \mu)^2}{2\sigma^2} \right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dlnorm`(*x: float*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → float

Density function of the Log-Normal distribution.

Example

```
>>> pystats.dlnorm(2.0, 1.0, 2.0)
0.0985685803440131
```

Parameters

- **x** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dlnorm**(*x: List[float]*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → List[float]

Density function of the Log-Normal distribution.

Example

```
>>> pystats.dlnorm([0.0, 1.0, 2.0], 1.0, 2.0)
[0.0, 0.17603266338214968, 0.0985685803440131]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the log-Normal distribution:

$$F(x;\mu,\sigma) = \int_0^x f(z;\mu,\sigma)dz = \frac{1}{2} + \frac{1}{2} \times \text{erf}\left(\frac{\ln(x)-\mu}{\sigma}\right)$$

where $\text{erf}(\cdot)$ denotes the Gaussian error function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**plnorm**(*p: float*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → float

Distribution function of the Log-Normal distribution.

Example

```
>>> pystats.plnorm(2.0, 1.0, 2.0)
0.43903100974768944
```

Parameters

- **p** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.plnorm`(*p: List[float]*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → List[float]
Distribution function of the Log-Normal distribution.

Example

```
>>> pystats.plnorm([0.0, 1.0, 2.0], 1.0, 2.0)
[0.0, 0.3085375387259869, 0.43903100974768944]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the log-Normal distribution:

$$q(p; \mu, \sigma) = \exp\left(\mu + \sqrt{2}\sigma \times \text{erf}^{-1}(2p - 1)\right)$$

where $\text{erf}^{-1}(\cdot)$ denotes the inverse Gaussian error function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qlnorm`(*q: float*, *mean: float = 0.0*, *sd: float = 1.0*) → float
Quantile function of the Log-Normal distribution.

Example

```
>>> pystats.qlnorm(0.95, 1.0, 2.0)
72.94511097708158
```

Parameters

- **q** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qlnorm**(*q: List[float]*, *mean: float = 0.0*, *sd: float = 1.0*) → List[float]
Quantile function of the Log-Normal distribution.

Example

```
>>> pystats.qlnorm([0.1, 0.5, 0.9], 1.0, 2.0)
[0.20948500212405705, 2.718281828459045, 35.27248263126183]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the log-Normal distribution is achieved by simulating $X \sim N(\mu, \sigma^2)$, then returning

$$Z = \exp(X) \sim \text{Lognormal}(\mu, \sigma^2)$$

Scalar Output

pystats.**rlnorm**(*mean: float = 0.0*, *sd: float = 1.0*) → float
Random sampling function for the Log-Normal distribution.

Example

```
>>> pystats.rlnorm(1.0, 2.0)
0.7961734447160091
```

Parameters

- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A pseudo-random draw from the Log-Normal distribution.

List Output

`pystats.rlnorm(n: int, mean: float = 0.0, sd: float = 1.0) → List[float]`

Random sampling function for the Log-Normal distribution.

Example

```
>>> pystats.rlnorm(3, 1.0, 2.0)
[0.7889982649469498, 0.060477695435514324, 0.091500401970679 03]
```

Parameters

- **n** (*int*) – The number of output values.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A list of pseudo-random draws from the Log-Normal distribution.

dlnorm	Density function of the log Normal distribution
plnorm	Distribution function of the log Normal distribution
qlnorm	Quantile function of the log Normal distribution
rlnorm	Sampling function of the log Normal distribution

2.3.12 Logistic Distribution

Table of contents

Density Function

The density function of the Logistic distribution:

$$f(x;\mu,\sigma) = \frac{\exp\left(-\frac{x-\mu}{\sigma}\right)}{\sigma\left(1+\exp\left(-\frac{x-\mu}{\sigma}\right)\right)^2}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dlogis`(*x: float*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → float

Density function of the Logistic distribution.

Example

```
>>> pystats.dlogis(2.0, 1.0, 2.0)
0.11750185610079714
```

Parameters

- **x** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dlogis`(*x: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]

Density function of the Logistic distribution.

Example

```
>>> pystats.dlogis([0.0, 1.0, 2.0], 1.0, 2.0)
[0.11750185610079714, 0.125, 0.11750185610079714]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Logistic distribution:

$$F(x;\mu,\sigma) = \int_{-\infty}^{x} f(z;\mu,\sigma)dz = \frac{1}{1+\exp\left(-\frac{x-\mu}{\sigma}\right)}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**plogis**(*p: float*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → float

Distribution function of the Logistic distribution.

Example

```
>>> pystats.plogis(2.0, 1.0, 2.0)
0.6224593312018546
```

Parameters

- **p** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

pystats.**plogis**(*p: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Logistic distribution.

Example

```
>>> pystats.plogis([0.0, 1.0, 2.0], 1.0, 2.0)
[0.37754066879814546, 0.5, 0.6224593312018546]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Logistic distribution:

$$q(p; \mu, \sigma) = \mu + \sigma \times \ln \left(\frac{p}{1-p} \right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qlogis**(*q: float*, *mu: float = 0.0*, *sigma: float = 1.0*) → float

Quantile function of the Logistic distribution.

Example

```
>>> pystats.qlogis(0.75, 1.0, 2.0)
3.1972245773362196
```

Parameters

- **q** (*float*) – A real-valued input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qlogis**(*q: List[float]*, *mu: float = 0.0*, *sigma: float = 1.0*) → List[float]

Quantile function of the Logistic distribution.

Example

```
>>> pystats.qlogis([0.1, 0.3, 0.7], 1.0, 2.0)
[-3.394449154672439, -0.6945957207744073, 2.694595720774407]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Logistic distribution is achieved via the inverse probability integral transform.

Scalar Output

pystats.**rlogis**(*mu: float = 0.0*, *sigma: float = 1.0*) → float

Random sampling function for the Logistic distribution.

Example

```
>>> pystats.rlogis(1.0, 2.0)
-2.0430312686217516
```

Parameters

- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A pseudo-random draw from the Logistic distribution.

List Output

pystats.**rlogis**(*n: int*, *mu: float = 0.0*, *sigma: float = 1.0*) → List[float]

Random sampling function for the Logistic distribution.

Example

```
>>> pystats.rlogis(3, 1.0, 2.0)
[7.012051380112511, 1.4135266403017916, -1.3985463825344762]
```

Parameters

- **n** (*int*) – The number of output values.
- **mu** (*float*) – The location parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of pseudo-random draws from the Logistic distribution.

dlogis	Density function of the Logistic distribution
plogis	Distribution function of the Logistic distribution
qlogis	Quantile function of the Logistic distribution
rlogis	Sampling function of the Logistic distribution

2.3.13 Normal Distribution

Table of contents

Density Function

The density function of the Normal (Gaussian) distribution:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dnorm`(*x: float*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → float

Density function of the Normal distribution.

Example

```
>>> pystats.dnorm(1.645, 0.0, 1.0)
0.10311081109198143
```

Parameters

- **x** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dnorm`(*x: List[float]*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → List[float]
Density function of the Normal distribution.

Example

```
>>> pystats.dnorm([-1.0, 0.0, 2.0], 0.0, 1.0)
[0.24197072451914342, 0.39894228040143265, 0.05399096651318805]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Normal (Gaussian) distribution:

$$F(x;\mu,\sigma) = \int_{-\infty}^{x} f(z;\mu,\sigma) dz = \frac{1}{2} \times \left(1 + \text{erf}\left(\frac{x-\mu}{\sqrt{2}\sigma} \right) \right)$$

where $\text{erf}(\cdot)$ denotes the Gaussian error function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pnorm`(*p: float*, *mean: float = 0.0*, *sd: float = 1.0*, *log: bool = False*) → float
Distribution function of the Normal distribution.

Example

```
>>> pystats.pnorm(1.645, 0.0, 1.0)
0.9500150944608786
```

Parameters

- **p** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.pnorm(p: List[float], mean: float = 0.0, sd: float = 1.0, log: bool = False) → List[float]`

Distribution function of the Normal distribution.

Example

```
>>> pystats.pnorm([-1.0, 0.0, 2.0], 0.0, 1.0)
[0.15865525393145705, 0.5, 0.9772498680518208]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the log-Normal distribution:

$$q(p; \mu, \sigma) = \mu + \sqrt{2}\sigma \times \text{erf}^{-1}(2p - 1)$$

where $\text{erf}^{-1}(\cdot)$ denotes the inverse Gaussian error function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qnorm(q: float, mean: float = 0.0, sd: float = 1.0) → float`

Quantile function of the Normal distribution.

Example

```
>>> pystats.qnorm(0.95, 0.0, 1.0)
1.6448536269514706
```

Parameters

- **q** (*float*) – A real-valued input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qnorm`(*q: List[float]*, *mean: float = 0.0*, *sd: float = 1.0*) → List[float]

Quantile function of the Normal distribution.

Example

```
>>> pystats.qnorm([0.3, 0.5, 0.95], 0.0, 1.0)
[-0.5244005127080409, 0.0, 1.6448536269514706]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Normal distribution is achieved via the `normal_distribution` class from the C++ standard library, imported from `<random>`.

Scalar Output

`pystats.rnorm`(*mean: float = 0.0*, *sd: float = 1.0*) → float

Random sampling function for the Normal distribution.

Example

```
>>> pystats.rnorm(0.0, 1.0)
0.2772437359624193
```

Parameters

- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A pseudo-random draw from the Normal distribution.

List Output

`pystats.rnorm(n: int, mean: float = 0.0, sd: float = 1.0) → List[float]`

Random sampling function for the Normal distribution.

Example

```
>>> pystats.rnorm(3, 0.0, 1.0)
[-0.04796221296447198, -2.165514623383353, 0.7219224921308472]
```

Parameters

- **n** (*int*) – The number of output values.
- **mean** (*float*) – The mean parameter, a real-valued input.
- **sd** (*float*) – The standard deviation parameter, a real-valued input.

Returns A list of pseudo-random draws from the Normal distribution.

dnorm	Density function of the Normal distribution
pnorm	Distribution function of the Normal distribution
qnorm	Quantile function of the Normal distribution
rnorm	Sampling function of the Normal distribution

2.3.14 Poisson Distribution

Table of contents

Density Function

The density function of the Poisson distribution:

$$f(x; \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!} \times \mathbf{1}[x \geq 0]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dpois`(*x: float*, *rate: float*, *log: bool = False*) → float

Density function of the Poisson distribution.

Example

```
>>> pystats.dpois(8, 10.0)
0.11259903214902026
```

Parameters

- **x** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dpois`(*x: List[float]*, *rate: float*, *log: bool = False*) → List[float]

Density function of the Poisson distribution.

Example

```
>>> pystats.dpois([2, 3, 4], 10.0)
[0.0022699964881242435, 0.007566654960414158, 0.01891663740103532]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Poisson distribution:

$$F(x;\lambda) = \sum_{z \leq x} f(z;\lambda) = \exp(-\lambda) \sum_{z \leq x} \frac{\lambda^z}{z!} \times \mathbf{1}[z \geq 0]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.`**`ppois`**(*p: float*, *rate: float*, *log: bool = False*) → float

Distribution function of the Poisson distribution.

Example

```
>>> pystats.ppois(8, 10.0)
0.3328196787507189
```

Parameters

- **p** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.`**`ppois`**(*p: List[float]*, *rate: float*, *log: bool = False*) → List[float]

Distribution function of the Poisson distribution.

Example

```
>>> pystats.ppois([2, 3, 4], 10.0)
[0.0027693957155115762, 0.010336050675925718, 0.029252688076961075]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Poisson distribution:

$$q(p; \lambda) = \inf \{x : p \leq F(x; \lambda)\}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qpois(q: float, rate: float) → float`

Quantile function of the Poisson distribution.

Example

```
>>> pystats.qpois(0.9, 10.0)
14.0
```

Parameters

- **q** (*float*) – A real-valued input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qpois(q: List[float], rate: float) → List[float]`

Quantile function of the Poisson distribution.

Example

```
>>> pystats.qpois([0.3, 0.5, 0.8], 10.0)
[8.0, 10.0, 13.0]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Scalar Output

pystats.**rpois**(*rate: float*) → float

Random sampling function for the Poisson distribution.

Example

```
>>> pystats.rpois(10.0)
7.0
```

Parameters **rate** (*float*) – The rate parameter, a real-valued input.

Returns A pseudo-random draw from the Poisson distribution.

List Output

pystats.**rpois**(*n: int*, *rate: float*) → List[float]

Random sampling function for the Poisson distribution.

Example

```
>>> pystats.rpois(4, 10.0)
[11.0, 8.0, 8.0, 9.0]
```

Parameters

- **n** (*int*) – The number of output values.
- **rate** (*float*) – The rate parameter, a real-valued input.

Returns A list of pseudo-random draws from the Poisson distribution.

dpois	Density function of the Poisson distribution
ppois	Distribution function of the Poisson distribution
qpois	Quantile function of the Poisson distribution
rpois	Sampling function of the Poisson distribution

2.3.15 Student's t-Distribution

Table of contents

Density Function

The density function of the Student's t distribution:

$$f(x;\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

where $\Gamma(\cdot)$ denotes the gamma function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dt`(*x: float*, *dof: float = 1.0*, *log: bool = False*) → float

Density function of the t-distribution distribution.

Example

```
>>> pystats.dt(0.37, 11)
0.362095719673348
```

Parameters

- **x** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

`pystats.dt`(*x: List[float]*, *dof: float = 1.0*, *log: bool = False*) → List[float]
Density function of the t-distribution distribution.

Example

```
>>> pystats.dt([1.8, 0.7, 4.2], 11)
[0.08286084841281988, 0.3002542334010353, 0.0012519045841828374]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Student's t distribution:

$$F(x;\nu) = \int_{-\infty}^{x} f(z;\nu) dz = \frac{1}{2} + x\Gamma\left(\frac{\nu+1}{2}\right) + \frac{{}_2F_1\left(\frac{1}{2}, \frac{\nu+1}{2}; \frac{3}{\nu}; -\frac{x^2}{\nu}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}$$

where $\Gamma(\cdot)$ denotes the gamma function and ${}_2F_1$ denotes the hypergeometric function.

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.pt`(*p: float*, *dof: float = 1.0*, *log: bool = False*) → float
Distribution function of the t-distribution distribution.

Example

```
>>> pystats.pt(0.37, 11)
0.6407962382848924
```

Parameters

- **p** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.pt(p: List[float], dof: float = 1.0, log: bool = False) → List[float]`

Distribution function of the t-distribution distribution.

Example

```
>>> pystats.pt([1.8, 0.7, 4.2], 11)
[0.9503420534306152, 0.7507677671528026, 0.9992572076935229]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Student's t distribution:

$$q(p; \nu) = \inf \{x : p \leq F(x; \nu)\}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qt(q: float, dof: float = 1.0) → float`

Quantile function of the t-distribution distribution.

Example

```
>>> pystats.qt(0.5, 11)
0.0
```

Parameters

- **q** (*float*) – A real-valued input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qt(q: List[float], dof: float = 1.0)` → List[float]

Quantile function of the t-distribution distribution.

Example

```
>>> pystats.qt([0.3, 0.5, 0.8], 11)
[-0.5399378774105429, 0.0, 0.8755299114635708]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Scalar Output

`pystats.rt(dof: float = 1.0)` → float

Random sampling function for the t-distribution distribution.

Example

```
>>> pystats.rt(dof=11)
0.30585871954611826
```

Parameters **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A pseudo-random draw from the t-distribution distribution.

List Output

`pystats.rt(n: int, dof: float = 1.0)` → List[float]

Random sampling function for the t-distribution distribution.

Example

```
>>> pystats.rt(3, 11)
[2.781125786671334, -0.7090818248684807, -0.3220992876319297]
```

Parameters

- **n** (*int*) – The number of output values.
- **dof** (*float*) – The degrees of freedom parameter, a real-valued input.

Returns A list of pseudo-random draws from the t-distribution distribution.

dt	Density function of the t-distribution
pt	Distribution function of the t-distribution
qt	Quantile function of the t-distribution
rt	Sampling function of the t-distribution

2.3.16 Uniform Distribution

Table of contents

Density Function

The density function of the Uniform distribution:

$$f(x; a, b) = \frac{1}{b-a} \times \mathbf{1}[a \leq x \leq b]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**dunif**(*x: float*, *min: float = 0.0*, *max: float = 1.0*, *log: bool = False*) → float
Density function of the Uniform distribution.

Example

```
>>> pystats.dunif(0.5, -1.0, 2.0)
0.3333333333333333
```

Parameters

- **x** (*float*) – A real-valued input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dunif**(*x: List[float]*, *min: float = 0.0*, *max: float = 1.0*, *log: bool = False*) → List[float]
Density function of the Uniform distribution.

Example

```
>>> pystats.dunif([-2.0, 0.0, 2.0], -1.0, 3.0)
[0.0, 0.25, 0.25]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Uniform distribution:

$$F(x; a, b) = \int_a^x f(z; a, b) dz = \frac{x - a}{b - a} \times \mathbf{1}[a \leq x \leq b] + \times \mathbf{1}[x > b]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.punif`(*p: float*, *min: float = 0.0*, *max: float = 1.0*, *log: bool = False*) → float

Distribution function of the Uniform distribution.

Example

```
>>> pystats.punif(0.5, -1.0, 2.0)
0.5
```

Parameters

- **p** (*float*) – A real-valued input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

`pystats.punif`(*p: List[float]*, *min: float = 0.0*, *max: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Uniform distribution.

Example

```
>>> pystats.punif([-2.0, 0.0, 2.0], -1.0, 3.0)
[0.0, 0.25, 0.75]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Uniform distribution:

$$q(p; a, b) = a + p(b - a)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.qunif`(*q: float*, *min: float = 0.0*, *max: float = 1.0*) → float
Quantile function of the Uniform distribution.

Example

```
>>> pystats.qunif(0.75, -1.0, 2.0)
1.25
```

Parameters

- **q** (*float*) – A real-valued input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

`pystats.qunif`(*q: List[float]*, *min: float = 0.0*, *max: float = 1.0*) → List[float]
Quantile function of the Uniform distribution.

Example

```
>>> pystats.qunif([0.3, 0.5, 0.9], -1.0, 3.0)
[0.2, 1.0, 2.6]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Uniform distribution is achieved via the `uniform_real_distribution` class from the C++ standard library, imported from `<random>`.

Scalar Output

pystats.**runif**(*min: float = 0.0*, *max: float = 1.0*) → float

Random sampling function for the Uniform distribution.

Example

```
>>> pystats.runif(-1.0, 2.0)
1.0599044629461352
```

Parameters

- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.

Returns A pseudo-random draw from the Uniform distribution.

List Output

pystats.**runif**(*n: int*, *min: float = 0.0*, *max: float = 1.0*) → List[float]

Random sampling function for the Uniform distribution.

Example

```
>>> pystats.runif(3, -1.0, 2.0)
[1.0552953561528309, 0.81837699091385, 0.17769962603787082]
```

Parameters

- **n** (*int*) – The number of output values.
- **min** (*float*) – The lower limit of the distribution, a real-valued input.
- **max** (*float*) – The upper limit of the distribution, a real-valued input.

Returns A list of pseudo-random draws from the Uniform distribution.

dunif	Density function of the Uniform distribution
punif	Distribution function of the Uniform distribution
qnorm	Quantile function of the Uniform distribution
runif	Sampling function of the Uniform distribution

2.3.17 Weibull Distribution

Table of contents

Density Function

The density function of the Weibull distribution:

$$f(x; k, \theta) = \frac{k}{\theta} \left(\frac{x}{\theta} \right)^{k-1} \exp \left(- \left(\frac{x}{\theta} \right)^k \right) \times \mathbf{1}[x \geq 0]$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

`pystats.dweibull`(*x: float*, *alpha: float = 1.0*, *sigma: float = 1.0*, *log: bool = False*) → float

Density function of the Weibull distribution.

Example

```
>>> pystats.dweibull(1.0, 2.0, 3.0)
0.19885318151430437
```

Parameters

- **x** (*float*) – A real-valued input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The density function evaluated at *x*.

List Input

pystats.**dweibull**(*x: List[float]*, *alpha: float = 1.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]
Density function of the Weibull distribution.

Example

```
>>> pystats.dweibull([1.8, 0.7, 4.2], 2.0, 3.0)
[0.27907053042841234, 0.14731284075281734, 0.1314678595263087]
```

Parameters

- **x** (*List[float]*) – A standard list input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of density values corresponding to the elements of *x*.

Cumulative Distribution Function

The cumulative distribution function (CDF) of the Weibull distribution:

$$F(x; k, \theta) = \int_0^x f(z; k, \theta) dz = 1 - \exp\left(- \left(\frac{x}{\theta} \right)^k \times \mathbf{1}[x \geq 0] \right)$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**pweibull**(*p: float*, *alpha: float = 1.0*, *sigma: float = 1.0*, *log: bool = False*) → float
Distribution function of the Weibull distribution.

Example

```
>>> pystats.pweibull(1.0, 2.0, 3.0)
0.1051606831856301
```

Parameters

- **p** (*float*) – A real-valued input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns The cumulative distribution function evaluated at *p*.

List Input

pystats.**pweibull**(*p: List[float]*, *alpha: float = 1.0*, *sigma: float = 1.0*, *log: bool = False*) → List[float]

Distribution function of the Weibull distribution.

Example

```
>>> pystats.pweibull([1.8, 0.7, 4.2], 2.0, 3.0)
[0.302323673928969, 0.052988880874744404, 0.859141579078955]
```

Parameters

- **p** (*List[float]*) – A standard list input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.
- **log** (*bool*) – Return the log-density or the true form.

Returns A list of CDF values corresponding to the elements of *p*.

Quantile Function

The quantile function of the Weibull distribution:

$$q(p; k, \theta) = \lambda \times (-\ln(1-p))^{1/k}$$

Methods for scalar input, as well as for list input, are listed below.

Scalar Input

pystats.**qweibull**(*q: float*, *alpha: float = 1.0*, *sigma: float = 1.0*) → float

Quantile function of the Weibull distribution.

Example

```
>>> pystats.qweibull(0.5, 2.0, 3.0)
2.497663833473093
```

Parameters

- **q** (*float*) – A real-valued input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns The quantile function evaluated at *q*.

List Input

pystats.**qweibull**(*q: List[float]*, *alpha: float = 1.0*, *sigma: float = 1.0*) → List[float]

Quantile function of the Weibull distribution.

Example

```
>>> pystats.qweibull([0.3, 0.5, 0.9], 2.0, 3.0)
[1.7916680762486648, 2.497663833473093, 4.552281388155439]
```

Parameters

- **q** (*List[float]*) – A standard list input.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of quantiles values corresponding to the elements of *q*.

Random Sampling

Random sampling for the Weibull distribution is achieved via the inverse probability integral transform.

Scalar Output

pystats.**rweibull**(*alpha: float = 1.0*, *sigma: float = 1.0*) → float

Random sampling function for the Weibull distribution.

Example

```
>>> pystats.rweibull(2.0, 3.0)
2.7238639049596536
```

Parameters

- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A pseudo-random draw from the Weibull distribution.

List Output

pystats.**rweibull**(*n: int*, *alpha: float = 1.0*, *sigma: float = 1.0*) → List[float]
Random sampling function for the Weibull distribution.

Example

```
>>> pystats.rweibull(3, 2.0, 3.0)
[0.563858221268503, 1.10144762159266, 2.0484747373540606]
```

Parameters

- **n** (*int*) – The number of output values.
- **alpha** (*float*) – The shape parameter, a real-valued input.
- **sigma** (*float*) – The scale parameter, a real-valued input.

Returns A list of pseudo-random draws from the Weibull distribution.

dweibull	Density function of the Weibull distribution
pweibull	Distribution function of the Weibull distribution
qweibull	Quantile function of the Weibull distribution
rweibull	Sampling function of the Weibull distribution

2.4 Other statistical functions

Table of contents

2.4.1 Mean

The mean of the input data is calculated using:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}$$

pystats.**mean**(*x: List[float]*) → float
Compute the mean of the given data

Example

```
>>> pystats.mean([1,2,3,4,5])
3.0
```

Parameters **x** (*List[float]*) – A standard list input.

Returns The mean of the values in *x*.

2.4.2 Standard deviation

The standard deviation of the input data is calculated using:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}}$$

This function returns the standard deviation of the sample; that is, it returns the unbiased standard deviation.

`pystats.sd`(*x: List[float]*) → float
Compute the standard deviation of the given data

Example

```
>>> pystats.sd([1,2,3,4,5])
1.581138
```

Parameters **x** (*List[float]*) – A standard list input.

Returns The standard deviation (unbiased) of the values in *x*.

2.4.3 Variance

The variance of the input data is calculated using:

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}$$

This function returns the variance of the sample; that is, it returns the unbiased variance.

`pystats.var`(*x: List[float]*) → float
Compute the variance of the given data

Example

```
>>> pystats.var([1,2,3,4,5])
2.5
```

Parameters **x** (*List[float]*) – A standard list input.

Returns The variance (unbiased) of the values in *x*.