
Ncephes

Release unknown

Mar 13, 2019

Table of contents

1	NCephes	1
1.1	Usage	1
1.2	Install	2
1.3	Running the tests	2
1.4	Authors	2
1.5	License	2
2	Probability functions	3
2.1	Beta distribution	3
2.2	Binomial distribution	5
2.3	Chi-square distribution	8
2.4	F distribution	10
2.5	Normal distribution	11

This package provides a python interface for the [Cephes](#) library. It also supports [Numba](#) and its `nopython` mode.

1.1 Usage

```
>>> from ncephes import incbet
>>> print("{:.3f}".format(incbet(1., 3., 0.3)))
0.657
```

You can also call them inside a numba function

```
>>> from ncephes import incbet
>>> from numba import jit
>>>
>>> @jit
... def numba_incbet(a, b, x):
...     return incbet(a, b, x)
>>>
>>> print("{:.3f}".format(numba_incbet(1., 3., 0.3)))
0.657
```

and with nopython mode and nogil enabled

```
>>> from ncephes import incbet
>>> from numba import jit
>>>
>>> @jit(nogil=True, nopython=True)
... def numba_incbet(a, b, x):
...     return incbet(a, b, x)
>>>
>>> print("{:.3f}".format(numba_incbet(1., 3., 0.3)))
0.657
```

One can also statically link the compiled Cephes libraries `ncprob` and `ncellf`. Please, have a peek at the `examples/prj_name` for a minimalistic example.

1.2 Install

The recommended way of installing it is via `conda`

```
conda install -c conda-forge ncephes
```

An alternative way would be via `pip`

```
pip install ncephes
```

1.3 Running the tests

After installation, you can test it

```
python -c "import ncephes; ncephes.test()"
```

as long as you have `pytest`.

1.4 Authors

- **Danilo Horta** - <https://github.com/Horta>

1.5 License

This project is licensed under the MIT License - see the [LICENSE](#) file for details

Probability integrals and their inverses.

2.1 Beta distribution

2.1.1 Cumulative distribution function

btddtr (*a*, *b*, *x*)

Returns the area from zero to *x* under the beta density function.

Parameters

- **a** (*float*) – a positive number
- **b** (*float*) – a positive number
- **x** (*float*) – any number within [0, 1]

See also *incbet* ().

Description

Returns the area from zero to *x* under the beta density function:

$$P(x \mid a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1} dt$$

This function is identical to the incomplete beta integral function *incbet* ().

The complemented function is:

$$1 - P(1-x \mid a, b) = \text{incbet}(b, a, x)$$

Accuracy

See `incbet()`.

Reference: <http://www.netlib.org/cephes/doubldoc.html#btdtr>

2.1.2 Incomplete beta function

incbet (*a*, *b*, *x*)

Returns incomplete beta integral of the arguments, evaluated from zero to *x*. The function is defined as

Parameters

- **a** (*float*) – a positive number
- **b** (*float*) – a positive number
- **x** (*float*) – any number within [0, 1]

See also `incbi()`.

Description

$$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1} dt$$

The domain of definition is $0 \leq x \leq 1$. In this implementation *a* and *b* are restricted to positive values. The integral from *x* to 1 may be obtained by the symmetry relation:

$$1 - \text{incbet}(a, b, x) = \text{incbet}(b, a, 1 - x)$$

The integral is evaluated by a continued fraction expansion or, when *b***x* is small, by a power series.

Accuracy

Tested at uniformly distributed random points (*a*, *b*, *x*) with *a* and *b* in “domain” and *x* between 0 and 1.

			Relative error	
arithmetic	domain	# trials	peak	rms
IEEE	0,5	10000	6.9e-15	4.5e-16
IEEE	0,85	250000	2.2e-13	1.7e-14
IEEE	0,1000	30000	5.3e-12	6.3e-13
IEEE	0,10000	250000	9.3e-11	7.1e-12
IEEE	0,100000	10000	8.7e-10	4.8e-11

Outputs smaller than the IEEE gradual underflow threshold were excluded from these statistics.

Error messages

message	condition	value returned
incbet domain	$x < 0, x > 1$	0.0
incbet underflow		0.0

Reference: <http://www.netlib.org/cephes/doubldoc.html#incbet>

2.1.3 Inverse of incomplete beta function

incbi (*a*, *b*, *y*)

Given *y*, the function finds *x* such that $\text{incbet}(a, b, y) = x$.

Parameters

- **a** (*float*) – a positive number
- **b** (*float*) – a positive number
- **x** (*float*) – any number within [0, 1]

See also `incbet()`.

Description

The routine performs interval halving or Newton iterations to find the root of $\text{incbet}(a, b, x) - y = 0$.

Accuracy

				relative error	
	x	a, b			
arithmetic	domain	domain	# trials	peak	rms
IEEE	0, 1	.5, 10000	50000	5.8e-12	1.3e-13
IEEE	0, 1	.25, 100	100000	1.8e-13	3.9e-15
IEEE	0, 1	0, 5	50000	1.1e-12	5.5e-15
VAX	0, 1	.5, 100	25000	3.5e-14	1.1e-15
With a and b constrained to half-integer or integer values					
IEEE	0, 1	.5, 10000	50000	5.8e-12	1.1e-13
IEEE	0, 1	.5, 100	100000	1.7e-14	7.9e-16
With a=.5, b constrained to half-integer or integer values					
IEEE	0, 1	.5, 10000	10000	8.3e-11	1.0e-11

Reference: <http://www.netlib.org/cephes/doubldoc.html#incbi>

2.2 Binomial distribution

2.2.1 Cumulative distribution function

bdtr (*k*, *n*, *p*)

Returns the sum of the terms 0 through *k* of the Binomial probability density. The function is defined as:

Parameters

- **k** (*int*) – number of successes within [0, *n*]
- **n** (*int*) – number of trials
- **p** (*float*) – probability of success within [0, 1]

See also `bdtrc()` and `bdtri()`.

Description

$$\sum_{j=0}^k \binom{n}{j} p^j (1-p)^{n-j}$$

The terms are not summed directly; instead the incomplete beta integral is employed, according to the formula:

$$y = \text{bdtr}(k, n, p) = \text{incbet}(n - k, k + 1, 1 - p)$$

The arguments must be positive, with p ranging from 0 to 1.

Accuracy

Tested at random points (a, b, p) , with p between 0 and 1.

	a, b		relative error	
arithmetic	domain	# trials	peak	rms
For p between 0.001 and 1				
IEEE	0, 100	100000	4.3e-15	2.6e-16

See also `incbi()`.

Error messages

message	condition	value returned
bdtr domain	$k < 0$	0.0
	$n < k$	
	$x < 0, x > 1$	

Reference: <http://www.netlib.org/cephes/doubldoc.html#bdtr>

2.2.2 Survival function

bdtrc (k, n, p)

Returns the sum of the terms $k + 1$ through n of the Binomial probability density:

Parameters

- **k** (*int*) – number of successes within $[0, n]$
- **n** (*int*) – number of trials
- **p** (*float*) – probability of success within $[0, 1]$

See also `bdtr()` and `bdtri()`.

Description

$$\sum_{j=k+1}^n \binom{n}{j} p^j (1-p)^{n-j}$$

The terms are not summed directly; instead the incomplete beta integral is employed, according to the formula:

```
y = bdtrc( k, n, p ) = incbet( k+1, n-k, p )
```

The arguments must be positive, with p ranging from 0 to 1.

Accuracy

Tested at random points (a, b, p).

	a, b		relative error	
arithmetic	domain	# trials	peak	rms
For p between 0.001 and 1				
IEEE	0, 100	100000	6.7e-15	8.2e-16
For p between 0 and .001				
IEEE	0, 100	100000	1.5e-13	2.7e-15

Error messages

message	condition	value returned
bdtrc domain	$x < 0, x > 1, n < k$	0.0

Reference: <http://www.netlib.org/cephes/doubldoc.html#bdtrc>

2.2.3 Inverse of the cumulative distribution function

bdtri (*k*, *n*, *y*)

Finds the event probability *p* such that the sum of the terms 0 through *k* of the Binomial probability density is equal to the given cumulative probability *y*.

Parameters

- **k** (*int*) – number of successes within [0, *n*]
- **n** (*int*) – number of trials
- **y** (*float*) – cumulative probability within [0, 1]

See also *bdtr* () and *bdtrc* () .

Description

This is accomplished using the inverse beta integral function and the relation:

```
1 - p = incbi(n - k, k + 1, y)
```

Accuracy

Tested at random points (a, b, p).

	a, b		relative error	
arithmetic	domain	# trials	peak	rms
For p between 0.001 and 1				
IEEE	0, 100	100000	2.3e-14	6.4e-16
IEEE	0, 10000	100000	6.6e-12	1.2e-13
For p between 10 ⁻⁶ and 0.001				
IEEE	0, 100	100000	2.0e-12	1.3e-14
IEEE	0, 10000	100000	1.5e-12	3.2e-14

See also `incbi()`.

Error messages

message	condition	value returned
bdtri domain	k < 0, n <= k	0.0
	x < 0, x > 1	

Reference: <http://www.netlib.org/cephes/doubldoc.html#bdtri>

2.3 Chi-square distribution

2.3.1 Cumulative distribution function

chdtr(*k*, *x*)

Returns the area under the left hand tail (from 0 to *x*) of the Chi square probability density function with *k* degrees of freedom.

Parameters

- **k** (*int*) – degrees of freedom
- **x** (*float*) – positive Chi square variable

Description

$$P(x | k) = \frac{1}{\Gamma(k/2)} \int_0^{x/2} t^{k/2-1} e^{-t} dt$$

The incomplete gamma integral is used according to the formula:

`chdtr(k, x) = igam(k/2, x/2)`

The arguments must both be positive.

Accuracy

See `igam()` for accuracy.

Error messages

message	condition	value returned
chdtr domain	$x < 0$ or $v < 1$	0

Reference: <http://www.netlib.org/cephes/doubldoc.html#chdtr>

2.3.2 Survival function

chdtrc (k, x)

Returns the area under the right hand tail (from x to infinity) of the Chi square probability density function with k degrees of freedom.

Parameters

- **k** (*int*) – degrees of freedom
- **x** (*float*) – positive Chi square variable

Description

The incomplete gamma integral is used according to the formula:

```
chdtr(k, x) = igamc(k/2, x/2)
```

The arguments must both be positive.

Accuracy

See `igamc()` for accuracy.

Error messages

message	condition	value returned
chdtrc domain	$x < 0$ or $v < 1$	0

Reference: <http://www.netlib.org/cephes/doubldoc.html#chdtrc>

2.3.3 Inverse of the survival function

chdtri (k, y)

Finds the Chi-square argument x such that the integral from x to infinity of the Chi-square density is equal to the given cumulative probability y .

Parameters

- **k** (*int*) – degrees of freedom
- **y** (*float*) – cumulative probability

Description

This is accomplished using the inverse gamma integral function and the relation:

$$x/2 = \text{igami}(k/2, y)$$

Accuracy

See `igami()` for accuracy.

Error messages

message	condition	value returned
chdtri domain	$y < 0$ or $y > 1$	0
	$k < 1$	

Reference: <http://www.netlib.org/cephes/doubldoc.html#chdtri>

2.4 F distribution

2.4.1 Cumulative distribution function

fdtr (*df1*, *df2*, *x*)

Returns the area from zero to *x* under the F density function (also known as Snedcor's density or the variance ratio density).

Parameters

- **df1** (*int*) – degrees of freedom
- **df2** (*int*) – degrees of freedom
- **x** (*float*) – positive F variable

Description

This is the density of $x = (u1/df1)/(u2/df2)$, where *u1* and *u2* are random variables having Chi square distributions with *df1* and *df2* degrees of freedom, respectively.

The incomplete beta integral is used according to the formula:

$$P(x) = \text{incbet}(df1/2, df2/2, (df1 * x / (df2 + df1 * x)))$$

The arguments *a* and *b* are greater than zero, and *x* is nonnegative.

Accuracy

Tested at random points (*a*, *b*, *x*).

	x	a, b		relative error	
arithmetic	domain	domain	# trials	peak	rms
IEEE	0, 1	0, 100	100000	9.8e-15	1.7e-15
IEEE	1, 5	0, 100	100000	6.5e-15	3.5e-16
IEEE	0, 1	1, 10000	100000	2.2e-11	3.3e-12
IEEE	1, 5	1, 10000	100000	1.1e-11	1.7e-13

See also `incbet()`.

Error messages

message	condition	value returned
fdtr domain	a<0, b<0, x<0	0

Reference: <http://www.netlib.org/cephes/doubldoc.html#fdtr>

2.4.2 Survival function

2.4.3 Inverse of the cumulative distribution function

2.5 Normal distribution

2.5.1 Complementary error function

erfc(*x*)

Computes $1 - \text{erf}(x)$ in a numerically stable way.

Parameters *x* (*float*) – a real scalar.

Description

$$1 - \text{erf}(x) = \text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2) dt$$

For small x , $\text{erfc}(x) = 1 - \text{erf}(x)$; otherwise rational approximations are computed.

A special function `exp2()` is used to suppress error amplification in computing $\exp(-x^2)$.

Accuracy

			Relative error	
arithmetic	domain	# trials	peak	rms
IEEE	0, 26.6417	30000	1.3e-15	2.2e-16

Error messages

message	condition	value returned
erfc underflow	$x > 9.231948545$ (DEC)	0.0

Reference: <http://www.netlib.org/cephes/doubldoc.html#erfc>

2.5.2 Cumulative distribution function

ndtr(x)

Returns the area under the Gaussian probability density function, integrated from minus infinity to x .

Parameters x (*float*) – a real scalar.

Description

Area under the curve:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^x \exp(-t^2/2) dt$$

Equivalently, we have:

$$\text{ndtr}(x) = (1 + \text{erf}(z)) / 2 = \text{erfc}(z) / 2$$

where $z = x/\sqrt{2}$. Computation is done via the functions *erf()* and *erfc()* with care to avoid error amplification in computing $\exp(-x^2)$.

Accuracy

	x		relative error	
arithmetic	domain	# trials	peak	rms
IEEE	-13, 0	30000	1.3e-15	2.2e-16

Error messages

message	condition	value returned
erfc underflow	$x > 37.519379347$	0.0

Reference: <http://www.netlib.org/cephes/doubldoc.html#ndtr>

2.5.3 Error function

erf(x)

Parameters x (*float*) – a real scalar.

Description

The integral is

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dt.$$

The magnitude of x is limited to 9.231948545 for DEC arithmetic; 1 or -1 is returned outside this range.

For $0 \leq |x| < 1$, $\operatorname{erf}(x) = x * P4(x ** 2) / Q5(x ** 2)$; otherwise $\operatorname{erf}(x) = 1 - \operatorname{erfc}(x)$.

Accuracy

			Relative error	
arithmetic	domain	# trials	peak	rms
DEC	0, 1	14000	4.7e-17	1.5e-17
IEEE	0, 1	30000	3.7e-16	1.0e-16

Reference: <http://www.netlib.org/cephes/doubldoc.html#erf>

2.5.4 Inverse of the cumulative distribution function

ndtri (y)

Returns the argument x for which the area under the Gaussian probability density function (integrated from minus infinity to x) is equal to y .

Parameters y (*float*) – area under the curve.

Description

For small arguments $0 < y < \exp(-2)$, the program computes $z = \sqrt{-2.0 * \log y}$; then the approximation is

$$x = z - \log(z)/z - (1/z)P(1/z)/Q(1/z).$$

There are two rational functions P/Q , one for $0 < y < \exp(-32)$ and the other for y up to $\exp(-2)$. For larger arguments, $w = y - 0.5$, and

$$x/\sqrt{2\pi} = w + w^3 R(w^2)/S(w^2).$$

Accuracy

			Relative error	
arithmetic	domain	# trials	peak	rms
DEC	0.125, 1	5500	9.5e-17	2.1e-17
DEC	6e-39, 0.135	3500	5.7e-17	1.3e-17
IEEE	0.125, 1	20000	7.2e-16	1.3e-16
IEEE	3e-308, 0.135	50000	4.6e-16	9.8e-17

Error messages

message	condition	value returned
ndtri domain	$x \leq 0$	-MAXNUM
ndtri domain	$x \geq 1$	MAXNUM

Reference: <http://www.netlib.org/cephes/doubldoc.html#ndtri>

Reference: <http://www.netlib.org/cephes/cprob.tgz>

B

bdtr() (built-in function), 5
bdtrc() (built-in function), 6
bdtri() (built-in function), 7
btdtr() (built-in function), 3

C

chdtr() (built-in function), 8
chdtrc() (built-in function), 9
chdtri() (built-in function), 9

E

erf() (built-in function), 12
erfc() (built-in function), 11

F

fdtr() (built-in function), 10

I

incbet() (built-in function), 4
incbi() (built-in function), 5

N

ndtr() (built-in function), 12
ndtri() (built-in function), 13