# Math

## Table of Contents

# Beta

## IMPORTS

ML_Core.Math |

## DESCRIPTIONS

**FUNCTION** **Beta**

| Beta |
|------|
| (REAL8 x, REAL8 y) |

Return the beta value of two positive real numbers, x and y

**PARAMETER** <u>x</u> ||| REAL8 — the value of the first number

**PARAMETER** <u>y</u> ||| REAL8 — the value of the second number

**RETURN** **REAL8** — the beta value

# Distributions

## IMPORTS

ML_Core.Constants | ML_Core.Math |

## DESCRIPTIONS

**MODULE** **Distributions**

| | Distributions |
|---|---|

No Documentation Found

**Children**

1. Normal_CDF : Cumulative Distribution of the standard normal distribution, the probability that a normal random variable will be smaller than x standard deviations above or below the mean

2. Normal_PPF : Normal Distribution Percentage Point Function

3. T_CDF : Students t distribution integral evaluated between negative infinity and x

4. T_PPF : Percentage point function for the T distribution

5. Chi2_CDF : The cumulative distribution function for the Chi Square distribution

6. Chi2_PPF : The Chi Squared PPF function

## **FUNCTION** Normal_CDF

| REAL8 | Normal_CDF |
|-------|------------|
| (REAL8 x) | |

Cumulative Distribution of the standard normal distribution, the probability that a normal random variable will be smaller than x standard deviations above or below the mean. Taken from C/C++ Mathematical Algorithms for Scientists and Engineers, n. Shammas, McGraw-Hill, 1995

**PARAMETER** **x** ||| REAL8 — the number of standard deviations

**RETURN** **REAL8** —

## **FUNCTION** Normal_PPF

| REAL8 | Normal_PPF |
|-------|------------|
| (REAL8 x) | |

Normal Distribution Percentage Point Function. Translated from C/C++ Mathematical Algorithms for Scientists and Engineers, N. Shammas, McGraw-Hill, 1995

**PARAMETER** **x** ||| REAL8 — probability

**RETURN** **REAL8** —

## **FUNCTION** T_CDF

| REAL8 | **T__CDF** |
|---|---|
| (REAL8 x, REAL8 df) | |

Students t distribution integral evaluated between negative infinity and x. Translated from NIST SEL DATAPAC Fortran TCDF.f source

**PARAMETER** <u>**x**</u> ||| REAL8 — value of the evaluation

**PARAMETER** <u>**df**</u> ||| REAL8 — degrees of freedom

**RETURN** **REAL8** —

---

# FUNCTION T_PPF

Distributions \

| REAL8 | **T__PPF** |
|---|---|
| (REAL8 x, REAL8 df) | |

Percentage point function for the T distribution. Translated from NIST SEL DATAPAC Fortran TPPF.f source

**PARAMETER** <u>**x**</u> ||| REAL8 — No Doc

**PARAMETER** <u>**df**</u> ||| REAL8 — No Doc

**RETURN** **REAL8** —

---

# FUNCTION Chi2_CDF

Distributions \

| REAL8 | **Chi2__CDF** |
|---|---|
| (REAL8 x, REAL8 df) | |

The cumulative distribution function for the Chi Square distribution. the CDF for the specfied degrees of freedom. Translated from the NIST SEL DATAPAC Fortran subroutine CHSCDF.

**PARAMETER** <u>x</u> ||| REAL8 — No Doc

**PARAMETER** <u>df</u> ||| REAL8 — No Doc

**RETURN** **REAL8** —

---

# **FUNCTION** Chi2\_PPF

Distributions \

| REAL8 | **Chi2\_PPF** |
|-------|---------------|
| (REAL8 x, REAL8 df) | |

The Chi Squared PPF function. Translated from the NIST SEL DATAPAC Fortran subroutine CHSPPF.

**PARAMETER** <u>x</u> ||| REAL8 — No Doc

**PARAMETER** <u>df</u> ||| REAL8 — No Doc

**RETURN** **REAL8** —

---

# DoubleFac

## DESCRIPTIONS

**EMBED** **DoubleFac**

| REAL8 | **DoubleFac** |
|-------|---------------|
| (INTEGER2 i) | |

The 'double' factorial is defined for ODD n and is the product of all the odd numbers up to and including that number. We are extending the meaning to even numbers to mean the product of the even numbers up to and including that number. Thus DoubleFac(8) = 8*6*4*2 We also defend against i < 2 (returning 1.0)

**PARAMETER** **i** ||| INTEGER2 — the value used in the calculation

**RETURN** **REAL8** — the factorial of the sequence, declining by 2

# Fac

## DESCRIPTIONS

**EMBED** **Fac**

| REAL8 | **Fac** |
|-------|---------|
| (UNSIGNED2 i) | |

Factorial function

**PARAMETER** **i** ||| UNSIGNED2 — the value used, (i)(i-1)(i-2)...(2)

**RETURN** **REAL8** — the factorial i!

# gamma

## DESCRIPTIONS

**EMBED** **gamma**

| REAL8 | **gamma** |
|-------|-----------|
| (REAL8 x) | |

Return the value of gamma function of real number x A wrapper for the standard C tgamma function.

**PARAMETER** **x** ||| REAL8 — the input x

**RETURN** **REAL8** — the value of GAMMA evaluated at x

# log\_gamma

## DESCRIPTIONS

**EMBED** **log\_gamma**

| REAL8 | log\_gamma |
|---|---|
| (REAL8 x) | |

Return the value of the log gamma function of the absolute value of X. A wrapper for the standard C lgamma function. Avoids the race condition found on some platforms by taking the absolute value of the of the input argument.

**PARAMETER** **x** ||| REAL8 — the input x

**RETURN** **REAL8** — the value of the log of the GAMMA evaluated at ABS(x)

# lowerGamma

## DESCRIPTIONS

**EMBED** **lowerGamma**

| REAL8 | lowerGamma |
|-------|------------|
| (REAL8 x, REAL8 y) | |

Return the lower incomplete gamma value of two real numbers, x and y

**PARAMETER** x ||| REAL8 — the value of the first number

**PARAMETER** y ||| REAL8 — the value of the second number

**RETURN** REAL8 — the lower incomplete gamma value

# NCK

## IMPORTS

ML_Core.Math |

## DESCRIPTIONS

### FUNCTION NCK

| REAL8 | NCK |
|-------|-----|
| (INTEGER2 N, INTEGER2 K) | |

No Documentation Found

**PARAMETER** **n** ||| INTEGER2 — No Doc

**PARAMETER** **k** ||| INTEGER2 — No Doc

**RETURN** **REAL8** —

# Poly

## DESCRIPTIONS

**EMBED** **Poly**

| REAL8 | **Poly** |
|-------|----------|
| (REAL8 x, SET OF REAL8 Coeffs) | |

Evaluate a polynomial from a set of co-effs. Co-effs 1 is assumed to be the HIGH order of the equation. Thus for ax^2+bx+c - the set would need to be Coef := [a,b,c];

**PARAMETER** **Coeffs** ||| SET ( REAL8 ) — a set of coefficients forthe polynomial. The ALL set is considered to be all zero values

**PARAMETER** **x** ||| REAL8 — the value of x in the polynomial

**RETURN** **REAL8** — value of the polynomial at x

# StirlingFormula

[Go Up](#)

## IMPORTS

ML_Core.Math | ML_Core.Constants |

## DESCRIPTIONS

**FUNCTION** **StirlingFormula**

| **StirlingFormula** |
|---|
| `(REAL x)` |

Stirling's formula

**PARAMETER** <u>**x**</u> ||| REAL8 — the point of evaluation

**RETURN** **REAL8** — evaluation result

# upperGamma

---

## DESCRIPTIONS

**EMBED** **upperGamma**

| REAL8 | upperGamma |
|-------|------------|
| (REAL8 x, REAL8 y) | |

Return the upper incomplete gamma value of two real numbers, x and y.

**PARAMETER** x ||| REAL8 — the value of the first number

**PARAMETER** y ||| REAL8 — the value of the second number

**RETURN** **REAL8** — the upper incomplete gamma value

---