

Search:

Go

Not logged in

Reference <random> normal_distribution

register

log in

C++

Information

Tutorials

Reference

Articles

Forum

Reference

C library:

Containers:

Input/Output:

Multi-threading:

Other:

<algorithm>

<bitset>

<chrono>

<codecvt>

<complex>

<exception>

<functional>

<initializer_list>

<iterator>

<limits>

<locale>

<memory>

<new>

<numeric>

<random>

<ratio>

<regex>

<stdexcept>

<string>

<system_error>

<tuple>

<typeindex>

<typeinfo>

<type_traits>

<utility>

<valarray>

<random>

distributions:

bernoulli_distribution

binomial_distribution

cauchy_distribution

chi_squared_distribution

discrete_distribution

exponential_distribution

extreme_value_distribution

fisher_f_distribution

gamma_distribution

geometric_distribution

lognormal_distribution

negative_binomial_distribution

normal_distribution

piecewise_constant_distribution

piecewise_linear_distribution

poisson_distribution

student_t_distribution

uniform_int_distribution

uniform_real_distribution

weibull_distribution

generators:

default_random_engine

discard_block_engine

independent_bits_engine

knuth_b

linear_congruential_engine

mersenne_twister_engine

minstd_rand

minstd_rand0

mt19937

mt19937_64

random_device

ranlux24

ranlux24_base

ranlux48

ranlux48_base

shuffle_order_engine

subtract_with_carry_engine

other:

generate_canonical

seed_seq

normal_distribution



class template <random>
std::normal_distribution

template <class RealType = double> class normal_distribution;
Normal distribution
Random number distribution that produces floating-point values according to a *normal distribution*, which is described by the following *probability density function*:

$$p(x|\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

This distribution produces random numbers around the distribution *mean* (μ) with a specific *standard deviation* (σ).
The normal distribution is a common distribution used for many kind of processes, since it is the distribution that the aggregation of a large number of independent random variables approximates to, when all follow the same distribution (no matter which distribution).
The distribution parameters, *mean* (μ) and *stddev* (σ), are set on *construction*.
To produce a random value following this distribution, call its member function *operator()*.

Template parameters

RealType
A floating-point type. Aliased as member type *result_type*.
By default, this is *double*.

Member types

The following aliases are member types of *normal_distribution*:

member type	definition	notes
<i>result_type</i>	The first template parameter (<i>RealType</i>)	The type of the numbers generated (defaults to <i>double</i>)
<i>param_type</i>	<i>not specified</i>	The type returned by member <i>param</i> .

Member functions

(constructor)	Construct normal distribution (public member function)
operator()	Generate random number (public member function)
reset	Reset distribution (public member function)
param	Distribution parameters (public member function)
min	Minimum value (public member function)
max	Maximum value (public member function)

Distribution parameters:

normal_distribution::mean	Distribution mean (public member function)
normal_distribution::stddev	Standard deviation (public member function)

Non-member functions

operator<<	Insert into output stream (function template)
operator>>	Extract from input stream (function template)
relational operators	Relational operators (function template)

Example

```
1 // normal_distribution
2 #include <iostream>
3 #include <random>
4
5 int main()
6 {
7     const int nrolls=10000; // number of experiments
8     const int nstars=100;   // maximum number of stars to distribute
9
10    std::default_random_engine generator;
11    std::normal_distribution<double> distribution(5.0,2.0);
12
13    int p[10]={};
14
15    for (int i=0; i<nrolls; ++i) {
16        double number = distribution(generator);
17        if ((number>=0.0)&&(number<10.0)) ++p[int(number)];
18    }
19
20    std::cout << "normal_distribution (5.0,2.0):" << std::endl;
21
```

normal_distribution::(constructor)

member functions:

normal_distribution::max

normal_distribution::mean

normal_distribution::min

normal_distribution::operator()

normal_distribution::param

normal_distribution::reset

normal_distribution::stddev

non-member functions:

operator<<

operator>>

relational operators

```
22 for (int i=0; i<10; ++i) {
23     std::cout << i << ". " << (i+1) << ": ";
24     std::cout << std::string(p[i]*nstars/nrolls, '*') << std::endl;
25 }
26
27 return 0;
28 }
```

Possible output:

```
normal_distribution (5.0,2.0):
0-1: *
1-2: ****
2-3: *****
3-4: *****
4-5: *****
5-6: *****
6-7: *****
7-8: *****
8-9: ****
9-10: *
```

Ultra Tune Car Servicing

Contact us for your servicing, tyre, vehicle maintenance and repair needs. Go to ultratune.com.au



See also

lognormal_distribution	Lognormal distribution (class template)
chi_squared_distribution	Chi-squared distribution (class template)
cauchy_distribution	Cauchy distribution (class template)
fisher_f_distribution	Fisher F-distribution (class template)
student_t_distribution	Student T-Distribution (class template)

[Home page](#) | [Privacy policy](#)

© cplusplus.com, 2000-2017 - All rights reserved - v3.1
[Spotted an error? contact us](#)