**Math**

[Wiki](https://github.com/mbostock/d3/wiki) ▸ API Reference ▸ Core ▸ **Math**

[**Math**](https://github.com/mbostock/d3/wiki/Math)

* [d3.random.normal](https://github.com/mbostock/d3/wiki/Math#wiki-random_normal) -生成具有正态分布规律的随机数。
* [d3.random.logNormal](https://github.com/mbostock/d3/wiki/Math#wiki-random_logNormal) -  生成具有对数正态分布规律的随机数。
* [d3.random.bates](https://github.com/mbostock/d3/wiki/Math#wiki-random_bates) -  生成具有贝茨分布规律的随机数 。
* [d3.random.irwinHall](https://github.com/mbostock/d3/wiki/Math#wiki-random_irwinHall) -  生成具有  Irwin–Hall 分布规律的随机数 。
* [d3.transform](https://github.com/mbostock/d3/wiki/Math#wiki-transform) - compute the standard form of a 2D matrix transform.

**伪随机数产生**

你可以使用内置函数[Math.random](https://developer.mozilla.org/en-US/docs/JavaScript/Reference/Global_Objects/Math/random)来生成统一的分布（uniform distributions）。例如，要生成介于0和99 （含）的随机整数，可以调Math.floor(Math.random() \* 100).

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-d3_random_normal) d3.random.**normal**([*mean*, [*deviation*]])

返回一个符合正态（高斯）分布[normal (Gaussian) distribution](http://en.wikipedia.org/wiki/Normal_distribution)的随机数.

随机变量的期望值是*mean（*默认为0.0*）*, 标准差是*deviation （*默认为1.0*）*.

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-d3_random_logNormal) d3.random.**logNormal**([*mean*, [*deviation*]])

返回一个满足对数分布 [log-normal distribution](http://en.wikipedia.org/wiki/Log-normal_distribution)的随机数.

随机变量自然对数的期望值是*mean（*默认为0.0*）*, 标准差是*deviation （*默认为1.0*）*.

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-d3_random_bates) d3.random.**bates**(*count*)

返回一个符合贝茨分布[Bates distribution](http://en.wikipedia.org/wiki/Bates_distribution)的随机数.*count*指定自变量的个数.

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-d3_random_irwinHall) d3.random.**irwinHall**(*count*)

返回一个 符合欧文霍尔分布 [Irwin–Hall distribution](http://en.wikipedia.org/wiki/Irwin%E2%80%93Hall_distribution)的随机数. *count*指定自变量的个数.

**2D 变换**

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-d3_transform) d3.**transform**(*string*)

解析给定的2D仿射变换字符串，依照SVG的变换属性[transform attribute](http://www.w3.org/TR/SVG/coords.html#TransformAttribute)的定义.

分解这个字符串为一个由 翻译translate, 旋转rotate, X偏移x-skew和尺度 scale组成的标准表示。此行为规范由CSS定义，参见：[matrix decomposition for animation](http://www.w3.org/TR/css3-2d-transforms/#matrix-decomposing).

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-transform_rotate) transform.**rotate**

返回此变换的旋转角θ ，以度为单位。

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-transform_translate) transform.**translate**

返回此变换的[ DX，DY ]翻译，局部坐标（通常为像素）的两元素数组

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-transform_skew) transform.**skew**

返回此变换的x的偏移φ ，单位为度。.

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-transform_scale) transform.**scale**

返回变换的[kx, ky] 尺度，一个两元素数组.

[#](file:///C:\Users\%E5%BC%A0%E5%A4%A9%E6%97%AD\AppData\Local\youdao\ynote\editor\web\editor.html#wiki-transform_toString) transform.**toString**

返回此转换的字符串表示形式，其形式为 "translate(dx,dy)rotate(θ)skewX(φ)scale(kx,ky)".

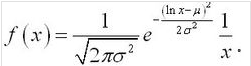
补充知识：

（1）对数正态分布（logarithmic normal distribution）

一个随机变量的对数服从正态分布，则该随机变量服从对数正态分布。

在概率论与统计学中，**对数正态分布**是对数为正态分布的任意随机变量的概率分布。**如果 *X* 是服从正态分布的随机变量，则 exp(*X*) 服从对数正态分布；同样，如果 *Y* 服从对数正态分布，则 ln(*Y*) 服从正态分布。 如果一个变量可以看作是许多很小独立因子的乘积，则这个变量可以看作是对数正态分布。一个典型的例子是股票投资的长期收益率，它可以看作是每天收益率的乘积。**

设ξ服从对数正态分布，其密度函数为：



数学期望和方差分别为：

C:\Users\张天旭\Desktop\2014-1-31 对数正态分布密度函数的数学期望和方差.png

理解：

要知道一个变量的值的概率分布就看这个变量的密度函数（直观的看就是看密度函数图像），

同时这个密度函数有自己的特征，用期望和方差来度量。

再理解：

变量的概率分布图像都是这样的：

横轴是变量的值，纵轴是取此值的概率（即每种变量值对应的概率和是1）

再理解：

股票长期收益率例子的理解：

股票长期收益率=每年的收益率乘积

每年的收益率不一样但是服从正态分布

（2）贝茨分布(Bates distribution)

（3）欧文霍尔分布(Irwin–Hall distribution)

（4）2D变换