# 随机抽样 (numpy.random)

**简单的随机数据**

|  |  |
| --- | --- |
| [rand](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.rand.html#numpy.random.rand)(d0, d1, ..., dn) | 随机值  >>> np.random.rand(3,2)  array([[ 0.14022471, 0.96360618], #random  [ 0.37601032, 0.25528411], #random  [ 0.49313049, 0.94909878]]) #random |
| [randn](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randn.html#numpy.random.randn)(d0, d1, ..., dn) | 返回一个样本，具有标准正态分布。  Notes  For random samples from , use:  sigma \* np.random.randn(...) + mu  Examples  >>> np.random.randn()  2.1923875335537315 #random  Two-by-four array of samples from N(3, 6.25):  >>> 2.5 \* np.random.randn(2, 4) + 3  array([[-4.49401501, 4.00950034, -1.81814867, 7.29718677], #random  [ 0.39924804, 4.68456316, 4.99394529, 4.84057254]]) #random |
| [randint](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randint.html#numpy.random.randint)(low[, high, size]) | 返回随机的整数，位于半开区间 [low, high)。  >>> np.random.randint(2, size=10)  array([1, 0, 0, 0, 1, 1, 0, 0, 1, 0])  >>> np.random.randint(1, size=10)  array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0])  Generate a 2 x 4 array of ints between 0 and 4, inclusive:  >>> np.random.randint(5, size=(2, 4))  array([[4, 0, 2, 1],  [3, 2, 2, 0]]) |
| [random\_integers](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.random_integers.html#numpy.random.random_integers)(low[, high, size]) | 返回随机的整数，位于闭区间 [low, high]。  Notes  To sample from N evenly spaced floating-point numbers between a and b, use:  a + (b - a) \* (np.random.random\_integers(N) - 1) / (N - 1.)  Examples  [复制代码](javascript:void(0);)  >>> np.random.random\_integers(5)  4  >>> type(np.random.random\_integers(5))  <type ‘int‘>  >>> np.random.random\_integers(5, size=(3.,2.))  array([[5, 4],  [3, 3],  [4, 5]])  [复制代码](javascript:void(0);)  Choose five random numbers from the set of five evenly-spaced numbers between 0 and 2.5, inclusive (*i.e.*, from the set ):  >>> 2.5 \* (np.random.random\_integers(5, size=(5,)) - 1) / 4.  array([ 0.625, 1.25 , 0.625, 0.625, 2.5 ])  Roll two six sided dice 1000 times and sum the results:  >>> d1 = np.random.random\_integers(1, 6, 1000)  >>> d2 = np.random.random\_integers(1, 6, 1000)  >>> dsums = d1 + d2  Display results as a histogram:  >>> import matplotlib.pyplot as plt  >>> count, bins, ignored = plt.hist(dsums, 11, normed=True)  >>> plt.show() |
| [random\_sample](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.random_sample.html#numpy.random.random_sample)([size]) | 返回随机的浮点数，在半开区间 [0.0, 1.0)。  To sample  multiply the output of [random\_sample](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.random_sample.html#numpy.random.random_sample) by *(b-a)* and add *a*:  (b - a) \* random\_sample() + a  Examples  [复制代码](javascript:void(0);)  >>> np.random.random\_sample()  0.47108547995356098  >>> type(np.random.random\_sample())  <type ‘float‘>  >>> np.random.random\_sample((5,))  array([ 0.30220482, 0.86820401, 0.1654503 , 0.11659149, 0.54323428])  [复制代码](javascript:void(0);)  Three-by-two array of random numbers from [-5, 0):  >>> 5 \* np.random.random\_sample((3, 2)) - 5  array([[-3.99149989, -0.52338984],  [-2.99091858, -0.79479508],  [-1.23204345, -1.75224494]]) |
| [random](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.random.html#numpy.random.random)([size]) | 返回随机的浮点数，在半开区间 [0.0, 1.0)。  （官网例子与random\_sample完全一样） |
| [ranf](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.ranf.html#numpy.random.ranf)([size]) | 返回随机的浮点数，在半开区间 [0.0, 1.0)。  （官网例子与random\_sample完全一样） |
| [sample](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.sample.html#numpy.random.sample)([size]) | 返回随机的浮点数，在半开区间 [0.0, 1.0)。  （官网例子与random\_sample完全一样） |
| [choice](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.choice.html#numpy.random.choice)(a[, size, replace, p]) | 生成一个随机样本，从一个给定的一维数组  Examples  Generate a uniform random sample from np.arange(5) of size 3:  >>> np.random.choice(5, 3)  array([0, 3, 4])  >>> #This is equivalent to np.random.randint(0,5,3)  Generate a non-uniform random sample from np.arange(5) of size 3:  >>> np.random.choice(5, 3, p=[0.1, 0, 0.3, 0.6, 0])  array([3, 3, 0])  Generate a uniform random sample from np.arange(5) of size 3 without replacement:  >>> np.random.choice(5, 3, replace=False)  array([3,1,0])  >>> #This is equivalent to np.random.permutation(np.arange(5))[:3]  Generate a non-uniform random sample from np.arange(5) of size 3 without replacement:  >>> np.random.choice(5, 3, replace=False, p=[0.1, 0, 0.3, 0.6, 0])  array([2, 3, 0])  Any of the above can be repeated with an arbitrary array-like instead of just integers. For instance:  [复制代码](javascript:void(0);)  >>> aa\_milne\_arr = [‘pooh‘, ‘rabbit‘, ‘piglet‘, ‘Christopher‘]  >>> np.random.choice(aa\_milne\_arr, 5, p=[0.5, 0.1, 0.1, 0.3])  array([‘pooh‘, ‘pooh‘, ‘pooh‘, ‘Christopher‘, ‘piglet‘],  dtype=‘|S11‘)  [复制代码](javascript:void(0);) |
| [bytes](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.bytes.html#numpy.random.bytes)(length) | 返回随机字节。  >>> np.random.bytes(10)  ‘ eh\x85\x022SZ\xbf\xa4‘ #random |

**排列**

|  |  |
| --- | --- |
| [shuffle](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.shuffle.html#numpy.random.shuffle)(x) | 现场修改序列，改变自身内容。（类似洗牌，打乱顺序）  >>> arr = np.arange(10)  >>> np.random.shuffle(arr)  >>> arr  [1 7 5 2 9 4 3 6 0 8]    This function only shuffles the array along the first index of a multi-dimensional array:  >>> arr = np.arange(9).reshape((3, 3))  >>> np.random.shuffle(arr)  >>> arr  array([[3, 4, 5],  [6, 7, 8],  [0, 1, 2]]) |
| [permutation](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.permutation.html#numpy.random.permutation)(x) | 返回一个随机排列  >>> np.random.permutation(10)  array([1, 7, 4, 3, 0, 9, 2, 5, 8, 6])  >>> np.random.permutation([1, 4, 9, 12, 15])  array([15, 1, 9, 4, 12])  >>> arr = np.arange(9).reshape((3, 3))  >>> np.random.permutation(arr)  array([[6, 7, 8],  [0, 1, 2],  [3, 4, 5]]) |

**分布**

|  |  |
| --- | --- |
| [beta](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.beta.html#numpy.random.beta)(a, b[, size]) | 贝塔分布样本，在 [0, 1]内。 |
| [binomial](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.binomial.html#numpy.random.binomial)(n, p[, size]) | 二项分布的样本。 |
| [chisquare](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.chisquare.html#numpy.random.chisquare)(df[, size]) | 卡方分布样本。 |
| [dirichlet](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.dirichlet.html#numpy.random.dirichlet)(alpha[, size]) | 狄利克雷分布样本。 |
| [exponential](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.exponential.html#numpy.random.exponential)([scale, size]) | 指数分布 |
| [f](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.f.html#numpy.random.f)(dfnum, dfden[, size]) | F分布样本。 |
| [gamma](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.gamma.html#numpy.random.gamma)(shape[, scale, size]) | 伽马分布 |
| [geometric](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.geometric.html#numpy.random.geometric)(p[, size]) | 几何分布 |
| [gumbel](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.gumbel.html#numpy.random.gumbel)([loc, scale, size]) | 耿贝尔分布。 |
| [hypergeometric](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.hypergeometric.html#numpy.random.hypergeometric)(ngood, nbad, nsample[, size]) | 超几何分布样本。 |
| [laplace](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.laplace.html#numpy.random.laplace)([loc, scale, size]) | 拉普拉斯或双指数分布样本 |
| [logistic](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.logistic.html#numpy.random.logistic)([loc, scale, size]) | Logistic分布样本 |
| [lognormal](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.lognormal.html#numpy.random.lognormal)([mean, sigma, size]) | 对数正态分布 |
| [logseries](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.logseries.html#numpy.random.logseries)(p[, size]) | 对数级数分布。 |
| [multinomial](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.multinomial.html#numpy.random.multinomial)(n, pvals[, size]) | 多项分布 |
| [multivariate\_normal](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.multivariate_normal.html#numpy.random.multivariate_normal)(mean, cov[, size]) | 多元正态分布。  >>> mean = [0,0]  >>> cov = [[1,0],[0,100]] # diagonal covariance, points lie on x or y-axis  >>> import matplotlib.pyplot as plt  >>> x, y = np.random.multivariate\_normal(mean, cov, 5000).T  >>> plt.plot(x, y, ‘x‘); plt.axis(‘equal‘); plt.show() |
| [negative\_binomial](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.negative_binomial.html#numpy.random.negative_binomial)(n, p[, size]) | 负二项分布 |
| [noncentral\_chisquare](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.noncentral_chisquare.html#numpy.random.noncentral_chisquare)(df, nonc[, size]) | 非中心卡方分布 |
| [noncentral\_f](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.noncentral_f.html#numpy.random.noncentral_f)(dfnum, dfden, nonc[, size]) | 非中心F分布 |
| [normal](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html#numpy.random.normal)([loc, scale, size]) | 正态(高斯)分布  Notes  The probability density for the Gaussian distribution is  where  is the mean and  the standard deviation. The square of the standard deviation, , is called the variance.  The function has its peak at the mean, and its “spread” increases with the standard deviation (the function reaches 0.607 times its maximum at  and [[R217]](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.normal.html#r217)).    Examples  Draw samples from the distribution:  >>> mu, sigma = 0, 0.1 # mean and standard deviation  >>> s = np.random.normal(mu, sigma, 1000)  Verify the mean and the variance:  >>> abs(mu - np.mean(s)) < 0.01  True  >>> abs(sigma - np.std(s, ddof=1)) < 0.01  True  Display the histogram of the samples, along with the probability density function:  [复制代码](javascript:void(0);)  >>> import matplotlib.pyplot as plt  >>> count, bins, ignored = plt.hist(s, 30, normed=True)  >>> plt.plot(bins, 1/(sigma \* np.sqrt(2 \* np.pi)) \*  ... np.exp( - (bins - mu)\*\*2 / (2 \* sigma\*\*2) ),  ... linewidth=2, color=‘r‘)  >>> plt.show()  [复制代码](javascript:void(0);) |
| [pareto](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.pareto.html#numpy.random.pareto)(a[, size]) | 帕累托（Lomax）分布 |
| [poisson](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.poisson.html#numpy.random.poisson)([lam, size]) | 泊松分布 |
| [power](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.power.html#numpy.random.power)(a[, size]) | Draws samples in [0, 1] from a power distribution with positive exponent a - 1. |
| [rayleigh](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.rayleigh.html#numpy.random.rayleigh)([scale, size]) | Rayleigh 分布 |
| [standard\_cauchy](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.standard_cauchy.html#numpy.random.standard_cauchy)([size]) | 标准柯西分布 |
| [standard\_exponential](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.standard_exponential.html#numpy.random.standard_exponential)([size]) | 标准的指数分布 |
| [standard\_gamma](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.standard_gamma.html#numpy.random.standard_gamma)(shape[, size]) | 标准伽马分布 |
| [standard\_normal](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.standard_normal.html#numpy.random.standard_normal)([size]) | 标准正态分布 (mean=0, stdev=1). |
| [standard\_t](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.standard_t.html#numpy.random.standard_t)(df[, size]) | Standard Student’s t distribution with df degrees of freedom. |
| [triangular](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.triangular.html#numpy.random.triangular)(left, mode, right[, size]) | 三角形分布 |
| [uniform](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.uniform.html#numpy.random.uniform)([low, high, size]) | 均匀分布 |
| [vonmises](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.vonmises.html#numpy.random.vonmises)(mu, kappa[, size]) | von Mises分布 |
| [wald](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.wald.html#numpy.random.wald)(mean, scale[, size]) | 瓦尔德（逆高斯）分布 |
| [weibull](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.weibull.html#numpy.random.weibull)(a[, size]) | Weibull 分布 |
| [zipf](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.zipf.html#numpy.random.zipf)(a[, size]) | 齐普夫分布 |

**随机数生成器**

|  |  |
| --- | --- |
| [RandomState](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.RandomState.html#numpy.random.RandomState) | Container for the Mersenne Twister pseudo-random number generator. |
| [seed](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.seed.html#numpy.random.seed)([seed]) | Seed the generator. |
| [get\_state](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.get_state.html#numpy.random.get_state)() | Return a tuple representing the internal state of the generator. |
| [set\_state](http://docs.scipy.org/doc/numpy/reference/generated/numpy.random.set_state.html#numpy.random.set_state)(state) | Set the internal state of the generator from a tuple. |