

SciPy.org (<http://scipy.org/>) Docs (<http://docs.scipy.org/>) SciPy v0.14.0 Reference Guide ([../index.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.gengamma.html))

Statistical functions (**scipy.stats**) ([../stats.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.html))

index ([../genindex.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.html))

modules ([../py-modindex.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.html))

next ([scipy.stats.genhalflogistic.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.genhalflogistic.html))

previous ([scipy.stats.gamma.html](http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.gamma.html))

scipy.stats.gengamma

scipy.stats.gengamma = <scipy.stats._continuous_distns.gengamma_gen object at 0x2b45d2fd27d0>
(http://github.com/scipy/scipy/blob/v0.14.0/scipy/stats/_continuous_distns.py) [\[source\]](#)

A generalized gamma continuous random variable.

Continuous random variables are defined from a standard form and may require some shape parameters to complete its specification. Any optional keyword parameters can be passed to the methods of the RV object as given below:

Parameters: **x** : *array_like*

quantiles

q : *array_like*

lower or upper tail probability

a, c : *array_like*

shape parameters

loc : *array_like, optional*

location parameter (default=0)

scale : *array_like, optional*

scale parameter (default=1)

size : *int or tuple of ints, optional*

shape of random variates (default computed from input arguments)

moments : *str, optional*

composed of letters ['mvsk'] specifying which moments to compute where 'm' = mean, 'v' = variance, 's' = (Fisher's) skew and 'k' = (Fisher's) kurtosis. (default='mv')

Alternatively, the object may be called (as a function) to fix the shape, location, and scale parameters returning a “frozen” continuous RV object:

rv = gengamma(a, c, loc=0, scale=1)

- Frozen RV object with the same methods but holding the given shape, location, and scale fixed.

Notes

The probability density function for gengamma is:

$$\text{gengamma.pdf}(x, a, c) = \text{abs}(c) * x^{(c*a-1)} * \exp(-x**c) / \text{gamma}(a)$$

for $x > 0$, $a > 0$, and $c \neq 0$.

Examples

```
>>> from scipy.stats import gengamma
>>> import matplotlib.pyplot as plt
>>> fig, ax = plt.subplots(1, 1)
```

>>>

Calculate a few first moments:

```
>>> a, c = 4.41623854294, 3.11930916792
>>> mean, var, skew, kurt = gengamma.stats(a, c, moments='mvsk')
```

Display the probability density function (pdf):

```
>>> x = np.linspace(gengamma.ppf(0.01, a, c),
...                 gengamma.ppf(0.99, a, c), 100)
>>> ax.plot(x, gengamma.pdf(x, a, c),
...         'r-', lw=5, alpha=0.6, label='gengamma pdf')
```

Alternatively, freeze the distribution and display the frozen pdf:

```
>>> rv = gengamma(a, c)
>>> ax.plot(x, rv.pdf(x), 'k-', lw=2, label='frozen pdf')
```

Check accuracy of cdf and ppf:

```
>>> vals = gengamma.ppf([0.001, 0.5, 0.999], a, c)
>>> np.allclose([0.001, 0.5, 0.999], gengamma.cdf(vals, a, c))
True
```

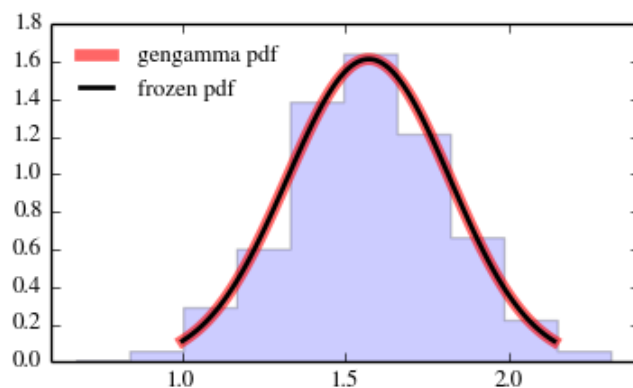
Generate random numbers:

```
>>> r = gengamma.rvs(a, c, size=1000)
```

And compare the histogram:

```
>>> ax.hist(r, normed=True, histtype='stepfilled', alpha=0.2)
>>> ax.legend(loc='best', frameon=False)
>>> plt.show()
```

(Source code ([./generated/scipy-stats-gengamma-1.py](#)))



Methods

<code>rvs(a, c, loc=0, scale=1, size=1)</code>	Random variates.
<code>pdf(x, a, c, loc=0, scale=1)</code>	Probability density function.
<code>logpdf(x, a, c, loc=0, scale=1)</code>	Log of the probability density function.
<code>cdf(x, a, c, loc=0, scale=1)</code>	Cumulative density function.
<code>logcdf(x, a, c, loc=0, scale=1)</code>	Log of the cumulative density function.
<code>sf(x, a, c, loc=0, scale=1)</code>	Survival function (1-cdf — sometimes more accurate).

`logsf(x, a, c, loc=0, scale=1)`

`ppf(q, a, c, loc=0, scale=1)`

`isf(q, a, c, loc=0, scale=1)`

`moment(n, a, c, loc=0, scale=1)`

`stats(a, c, loc=0, scale=1, moments='mv')`

`entropy(a, c, loc=0, scale=1)`

`fit(data, a, c, loc=0, scale=1)`

`expect(func, a, c, loc=0, scale=1, lb=None, ub=None, conditional=False, **kwds)`

`median(a, c, loc=0, scale=1)`

`mean(a, c, loc=0, scale=1)`

`var(a, c, loc=0, scale=1)`

`std(a, c, loc=0, scale=1)`

`interval(alpha, a, c, loc=0, scale=1)`

Log of the survival function.

Percent point function (inverse of cdf — percentiles).

Inverse survival function (inverse of sf).

Non-central moment of order n

Mean('m'), variance('v'), skew('s'), and/or kurtosis('k').

(Differential) entropy of the RV.

Parameter estimates for generic data.

Expected value of a function (of one argument) with respect to the distribution.

Median of the distribution.

Mean of the distribution.

Variance of the distribution.

Standard deviation of the distribution.

Endpoints of the range that contains alpha percent of the distribution

Previous topic

[scipy.stats.gamma \(scipy.stats.gamma.html\)](#)

Next topic

[scipy.stats.genhalflogistic \(scipy.stats.genhalflogistic.html\)](#)