scipy.stats.gamma

scipy.stats.gamma(**args*, ***kwds*) = *<scipy.stats._continuous_distns.gamma_gen object>* (https://github.com/scipy/scipy/blob/v1.6.0/scipy/stats/_continuous_distns.py) [source]

A gamma continuous random variable.

As an instance of the rv_continuous (scipy.stats.rv_continuous.html#scipy.stats.rv_continuous) class, gamma object inherits from it a collection of generic methods (see below for the full list), and completes them with details specific for this particular distribution.

See also:
erlang (scipy.stats.erlang.html#scipy.stats.erlang), expon (scipy.stats.expon.html#scipy.stats.expon)

Notes

The probability density function for gamma is:

$$f(x, a) = \frac{x^{a-1} e^{-x}}{\Gamma(a)}$$

for $x \geq 0$, $a > 0$. Here $\Gamma(a)$ refers to the gamma function.

gamma takes a as a shape parameter for a.

When a is an integer, gamma reduces to the Erlang distribution, and when $a = 1$ to the exponential distribution.

Gamma distributions are sometimes parameterized with two variables, with a probability density function of:

$$f(x, \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

Note that this parameterization is equivalent to the above, with `scale = 1 / beta`.

The probability density above is defined in the “standardized” form. To shift and/or scale the distribution use the `loc` and `scale` parameters. Specifically, `gamma.pdf(x, a, loc, scale)` is identically equivalent to `gamma.pdf(y, a) / scale` with `y = (x - loc) / scale`. Note that shifting the location of a distribution does not make it a “noncentral” distribution; noncentral generalizations of some distributions are available in separate classes.

Examples

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

Calculate a few first moments:

```
>>> a = 1.99
>>> mean, var, skew, kurt = gamma.stats(a, moments='mvsk')
```

Display the probability density function (`pdf`):

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

Alternatively, the distribution object can be called (as a function) to fix the shape, location and scale parameters. This returns a "frozen" RV object holding the given parameters fixed.

Freeze the distribution and display the frozen `pdf`:

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

Check accuracy of `cdf` and `ppf`:

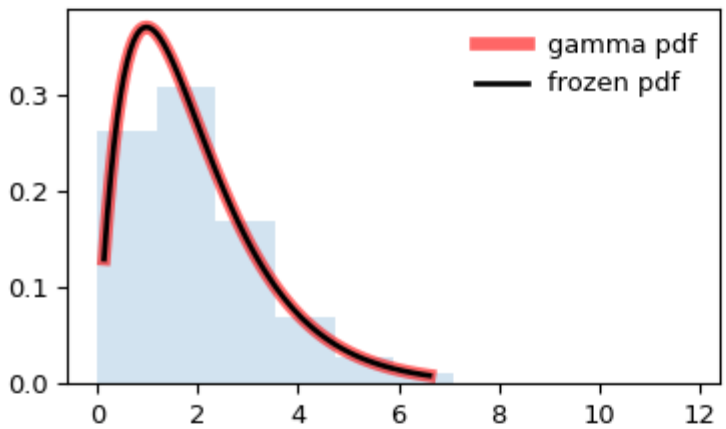
```
>>> vals = gamma.ppf([0.001, 0.5, 0.999], a)
>>> np.allclose([0.001, 0.5, 0.999], gamma.cdf(vals, a))
True
```

Generate random numbers:

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

And compare the histogram:

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



Methods

rvs(a, loc=0, scale=1, size=1, random_state=None)	Random variates.
pdf(x, a, loc=0, scale=1)	Probability density function.
logpdf(x, a, loc=0, scale=1)	Log of the probability density function.
cdf(x, a, loc=0, scale=1)	Cumulative distribution function.
logcdf(x, a, loc=0, scale=1)	Log of the cumulative distribution function.
sf(x, a, loc=0, scale=1)	Survival function (also defined as `1 - cdf`, but *sf* is sometimes more accurate).
logsf(x, a, loc=0, scale=1)	Log of the survival function.
ppf(q, a, loc=0, scale=1)	Percent point function (inverse of `cdf` — percentiles).
isf(q, a, loc=0, scale=1)	Inverse survival function (inverse of `sf`).
moment(n, a, loc=0, scale=1)	Non-central moment of order n
stats(a, loc=0, scale=1, moments='mv')	Mean('m'), variance('v'), skew('s'), and/or kurtosis('k').
entropy(a, loc=0, scale=1)	(Differential) entropy of the RV.
fit(data)	Parameter estimates for generic data. See scipy.stats.rv_continuous.fit (https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.rv_continuous.fit.html#scipy.stats.rv_continuous.fit) for detailed documentation of the keyword arguments.
expect(func, args=(a,), loc=0, scale=1, lb=None, ub=None, conditional=False, **kwds)	Expected value of a function (of one argument) with respect to the distribution.
median(a, loc=0, scale=1)	Median of the distribution.
mean(a, loc=0, scale=1)	Mean of the distribution.
var(a, loc=0, scale=1)	Variance of the distribution.
std(a, loc=0, scale=1)	Standard deviation of the distribution.
interval(alpha, a, loc=0, scale=1)	Endpoints of the range that contains alpha percent of the distribution

Previous topic

scipy.stats.gausshyper (scipy.stats.gausshyper.html)

Next topic

scipy.stats.gengamma (scipy.stats.gengamma.html)

Quick search

search