Documentation for `stats_arrays`

The `stats_arrays` package provides a standard NumPy array interface for defining uncertain parameters used in models, and classes for Monte Carlo sampling. It also plays well with others.

Motivation

- Want a consistent interface to SciPy and NumPy statistical function
- Want to be able to quickly load and save many parameter uncertainty distribution definitions in a portable format
- Want to manipulate and switch parameter uncertainty distributions and variables
- Want simple Monte Carlo random number generators that return a vector of parameter values to be fed into uncertainty or sensitivity analysis
- Want something simple, extensible, documented and tested

The `stats_arrays` package was originally developed for the Brightway2 life cycle assessment framework, but can be applied to any stochastic model.

Example

```
>>> from stats_arrays import *
>>> my_variables = UncertaintyBase.from_dicts(
...     {'loc': 2, 'scale': 0.5, 'uncertainty_type': NormalUncertainty.id},
...     {'loc': 1.5, 'minimum': 0, 'maximum': 10, 'uncertainty_type': TriangularUnce
... )
>>> my_variables
array([(2.0, 0.5, nan, nan, nan, False, 3),
       (1.5, nan, nan, 0.0, 10.0, False, 5)],
    dtype=[('loc', '<f8'), ('scale', '<f8'), ('shape', '<f8'),
           ('minimum', '<f8'), ('maximum', '<f8'), ('negative', '?'),
           ('uncertainty_type', 'u1')])
>>> my_rng = MCRandomNumberGenerator(my_variables)
>>> my_rng.next()
array([ 2.74414022,  3.54748507])
>>> # can also be used as an interator
>>> zip(my_rng, xrange(10))
[(array([ 2.96893108,  2.90654471]), 0),
 (array([ 2.31190619,  1.49471845]), 1),
 (array([ 3.02026168,  3.33696367]), 2),
 (array([ 2.04775418,  3.68356226]), 3),
 (array([ 2.61976694,  7.0149952 ]), 4),
 (array([ 1.79914025,  6.55264372]), 5),
 (array([ 2.2389968 ,  1.11165296]), 6),
 (array([ 1.69236527,  3.24463981]), 7),
 (array([ 1.77750176,  1.90119991]), 8),
 (array([ 2.32664152,  0.84490754]), 9)]
```

See a more complete notebook example.

Parameter array

The core data structure for `stats_arrays` is a parameter array, which is made from a special kind of NumPy array called a NumPy structured array which has the following data type:

```
import numpy as np
base_dtype = [
    ('loc', np.float64),
    ('scale', np.float64),
    ('shape', np.float64),
    ('minimum', np.float64),
```

```
    ('maximum', np.float64),
    ('negative', np.bool)
]
```

> **Note:**
>
> Read more on NumPy data types.

> **Note:**
>
> The *negative* column is used for uncertain parameters whose distributions are normally always positive, such as the lognormal, but in this case have negative values.

In general, most uncertainty distributions can be defined by three variables, commonly called *location*, *scale*, and *shape*. The *minimum* and *maximum* values make distributions **bounded**, so that one can, for example, define a normal uncertainty which is always positive.

> **Warning:**
>
> Bounds are not applied in the following methods: 1) Distribution functions (`PDF`, `CDF`, etc.) where you supply the input vector. 2) `.statistics`, which gives 95 percent confidence intervals for the unbounded distribution.

Heterogeneous parameter array

Parameter arrays can have multiple uncertainty distributions. To distinguish between the different distributions, another column, called `uncertainty_type`, is added:

```
heterogeneous_dtype = [
    ('uncertainty_type', np.uint8),
    ('loc', np.float64),
    ('scale', np.float64),
    ('shape', np.float64),
    ('minimum', np.float64),
    ('maximum', np.float64),
    ('negative', np.bool)
]
```

Note that *stats_arrays* was developed in conjunction with the Brightway LCA framework; Brightway uses the field name “uncertainty type”, without the underscore. Be sure to use the underscore when using *stats_arrays*.

Each uncertainty distribution has an integer ID number. See the table below for built-in distribution IDs.

> **Note:**
>
> The recommended way to use uncertainty distribution IDs is not by looking up the integers manually, but by referring to `SomeClass.id`, e.g. `LognormalDistribution.id`.

Mapping parameter array columns to uncertainty distributions

Name	ID	loc	scale	shape	minimum	maximum

Name	ID	`loc`	`scale`	`shape`	`minimum`	`maximum`
Undefined	0	**static value**				
No uncertainty	1	**static value**				
Lognormal [1]	2	μ	σ		*lower bound*	*upper bound*
Normal [2]	3	μ	σ		*lower bound*	*upper bound*
Uniform [3]	4				*minimum* [4]	**maximum**
Triangular [5]	5	*mode* [6]			*minimum* [7]	**maximum**
Bernoulli [8]	6	**p**			*lower bound*	*upper bound*
Discrete Uniform [9]	7				*minimum* [10]	**upper bound** [11]
Weibull [12]	8	*offset* [13]	λ	k		
Gamma [14]	9	*offset* [15]	θ	k		
Beta [16]	10	α	*upper bound*	β		
Generalized Extreme Value [17]	11	μ	σ	ξ		
Student's T [18]	12	*median*	*scale*	ν		

Items in **bold** are required, items in *italics* are optional.

[1] Lognormal distribution. μ and σ are the mean and standard deviation of the underlying normal distribution

[2] Normal distribution

[3] Uniform distribution

[4] Default is 0 if not otherwise specified

[5] Triangular distribution

[6] Default is $(minimum + maximum)/2$

[7] Default is 0 if not otherwise specified

[8] Bernoulli distribution. If `minimum` **and** `maximum` are specified, p is not limited to $0 < p < 1$, but instead to the interval $(minimum, maximum)$

[9] Discrete uniform

[10] The discrete uniform operates on a "half-open" interval, $[minimum, maximum)$, where the minimum is included but the maximum is not. Default is 0 if not otherwise specified.

[11] The distribution includes values up to, but not including, the `maximum`.

[12] Weibull distribution

[13] Optional offset from the origin

[14] Gamma distribution

[15] Optional offset from the origin

[16] Beta distribution

[17] Extreme value distribution

[18] Student's T distribution

Unused columns can be given any value, but it is recommended that they are set to `np.NaN`.

> **Warning:**
>
> Unused optional columns **must** be set to `np.NaN` to avoid unexpected behaviour!

Extending parameter arrays

Parameter arrays can have additional columns. For example, model parameters that will be inserted into a matrix could have columns called *row* and *column*. For speed reasons, it is recommended that only NumPy numeric types are used if the arrays are to stored on disk.

Technical reference

Probability distributions

- UncertaintyBase
- Lognormal
- Normal
- Uniform
- Discrete Uniform
- Triangular
- Bernoulli
- Beta
- Generalized Extreme Value
- Student's T
- Gamma
- Weibull

Random number generators

- Random number generator
- Monte Carlo random number generator
- Latin Hypercubic sampling

Indices and tables

- Index
- Module Index
- Search Page