Distributions and draws

HALF-NORMAL DENSITY

$$f_+(x) = 2\varphi(x) \quad (x \geq 0)$$

COMPLETE EXAMPLE

Scalar and vector draws

```
1  flatppl_compat = "0.1"
2
3  # Distribution constructors and variates.
4  standard = Normal(0, 1)
5  shifted = Normal(mu = 2, sigma = 0.5)
6  x ~ standard
7  y ~ shifted
8
9  # Array-valued product measures.
10 n = 3
11 means = [0.0, 1.0, 2.0]
12 u ~ iid(standard, n)
13 v ~ Normal.(mu = means, sigma = 0.5)
14
15 # Support restriction and normalization.
16 positive = normalize(truncate(
17     standard, interval(0, inf)))
18 z ~ positive
19
20 # Laws of scalar and array values.
21 scalar_law = lawof(x)
22 array_law = lawof(v)
```

A correlated vector

```
cov = rowstack([[1.0, 0.4], [0.4, 2.0]])
x_vec ~ MvNormal([0.0, 0.0], cov)
```

`MvNormal` takes a mean vector and a positive definite covariance matrix. Here the off-diagonal entry gives the covariance of the coordinates.

A random probability vector

```
p ~ Dirichlet([2.0, 2.0, 2.0])
category ~ Categorical(p)
counts ~ Multinomial(12, p)
```

Supply positive concentrations. The components of `p` sum to one. The category takes values 1, 2 or 3, and the counts sum to 12. Use `Categorical0` for labels 0, 1 and 2.

Distribution parameters by name

```
args = record(mu = 2.0, sigma = 0.5)
M = Normal(args)
```

Pass the constructor arguments as named record fields. The field `sigma` denotes the standard deviation.

ANNOTATED LINES

A Probability measures and variates 4–7

```
standard = Normal(0, 1)
shifted = Normal(mu = 2, sigma = 0.5)
x ~ standard
y ~ shifted
```

`Normal` constructs a probability measure. Name it with `=`. The declaration `x ~ standard` introduces a variate with that law. Constructors accept positional or named arguments.

B Independent arrays 10–13

```
n = 3
means = [0.0, 1.0, 2.0]
u ~ iid(standard, n)
v ~ Normal.(mu = means, sigma = 0.5)
```

`iid` uses the same law for all three entries. The dotted constructor pairs each entry of `means` with the common scale. Both produce a product measure on length-three arrays.

C Restrict the support 16–18

```
positive = normalize(truncate(
    standard, interval(0, inf)))
z ~ positive
```

`truncate` retains the mass on the specified set, here $\frac{1}{2}$. To draw from the conditional law, apply `normalize`. The retained mass must be finite and strictly positive.

D Laws of scalars and vectors 21–22

```
scalar_law = lawof(x)
array_law = lawof(v)
```

`lawof(x)` gives the scalar law. `lawof(v)` preserves the joint law of the vector. Its coordinates are independent in this example.

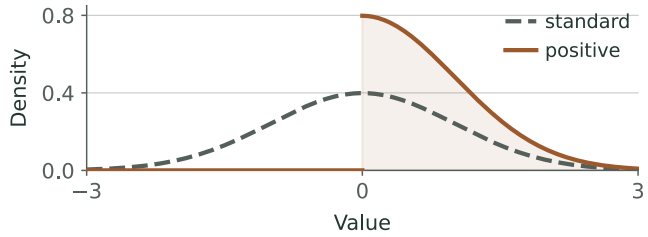
Mixture weights

```
mix = superpose(weighted(0.3, standard),
    weighted(0.7, shifted))
w ~ mix
```

A mixture with weights summing to one produces a single real draw. A draw from the product on line 13 has three coordinates.

CONSTRUCTOR INDEX

Restricting a standard Normal



The dashed curve has half its mass on either side of zero. The solid curve is the law used by `z` on line 18. Both integrate to one.

Location and scale

`Normal(mu, sigma)` $\mathbb{R}$
Mean and standard deviation.

`Cauchy(location, scale)` $\mathbb{R}$
Positive scale.

`StudentT(nu)` $\mathbb{R}$
Degrees of freedom. Use `locscale` for location and scale.

Positive and bounded values

`Gamma(shape, rate)` $x \geq 0$
`rate` is inverse scale, also in `Exponential(rate)`.

`InverseGamma(shape, scale)` $x > 0$
The second argument is the scale.

`LogNormal(mu, sigma)` $x > 0$
Mean and standard deviation of `log(x)`.

`Beta(alpha, beta)` $[0, 1]$
Two positive shape parameters.

`Uniform(interval(a, b))` $[a, b]$
Use finite endpoints with $a < b$. Use `joint` for a product of interval laws.

Counts and trials

Law	Count
`Bernoulli(p)`	One binary outcome
`Binomial(n,p)`	Successes in `n` trials
`Geometric(p)`	Failures before the first success

`Poisson(rate)` has mean and variance `rate`. For `NegativeBinomial2(mu,psi)` the mean is `mu` and the variance is $\mu + \frac{\mu^2}{\psi}$.

Jessica Cox · bcox@mpp.mpg.de