

## Re: random numbers according to user defined distribution ??

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2008-08/msg00654.html>

---

- *From:* Raymond Hettinger <python@xxxxxxx>
  - *Date:* Thu, 7 Aug 2008 02:13:53 -0700 (PDT)
- 

On Aug 6, 3:02 pm, Alex <axel.kow...@xxxxxxx> wrote:

I wonder if it is possible in python to produce random numbers according to a user defined distribution?  
Unfortunately the random module does not contain the distribution I need :-(

Sure there's a way but it won't be very efficient. Starting with an arbitrary probability density function over some range, you can run it through a quadrature routine to create a cumulative density function over that range. Use `random.random()` to create a uniform variate `x`. Then use a bisection search to find `x` in the cumulative density function over the given range.

```
from __future__ import division
from random import random

def integrate(f, lo, hi, steps=1000):
    dx = (hi - lo) / steps
    lo += dx / 2
    return sum(f(i*dx + lo) * dx for i in range(steps))

def make_cdf(f, lo, hi, steps=1000):
    total_area = integrate(f, lo, hi, steps)
    def cdf(x):
        assert lo <= x <= hi
        return integrate(f, lo, x, steps) / total_area
    return cdf

def bisect(target, f, lo, hi, n=20):
    'Find x between lo and hi where f(x)=target'
    for i in range(n):
        mid = (hi + lo) / 2.0
        if target < f(mid):
            hi = mid
        else:
            lo = mid
```

Re: random numbers according to user defined distribution ??

```
return (hi + lo) / 2.0
```

```
def make_user_distribution(f, lo, hi, steps=1000, n=20):
```

```
    cdf = make_cdf(f, lo, hi, steps)
```

```
    return lambda: bisect(random(), cdf, lo, hi, n)
```

```
if __name__ == '__main__':
```

```
    def linear(x):
```

```
        return 3 * x - 6
```

```
    lo, hi = 2, 10
```

```
    r = make_user_distribution(linear, lo, hi)
```

```
    for i in range(20):
```

```
        print r()
```

```
--
```

Raymond

.