

# Inverse CDF Sampling

Troy Stribling

July 4, 2018

## 1 Introduction

Inverse [CDF](#) sampling is a method for obtaining samples from both discrete and continuous probability distributions that requires the CDF to be invertable. The method assumes values of the CDF are Uniform random variables on  $[0, 1]$ . Values are generated and used as input into the inverted CDF to obtain samples with the distribution defined by the CDF. Here examples for both cases are discussed. For the continuous case a proof is given that demonstrates the samples produced have the expected distribution.

## 2 Sampling Discrete Distributions

A discrete probability distribution consisting of a finite set of  $N$  probability values is defined by,  $\{p_1, p_2, \dots, p_N\}$  with  $p_i \geq 0, \forall i$  and  $\sum_{i=1}^N p_i = 1$ .

The CDF specifies the probability that  $i \leq n$  and is given by,

$$P(i \leq n) = P(n) = \sum_{i=1}^n p_i, \quad (1)$$

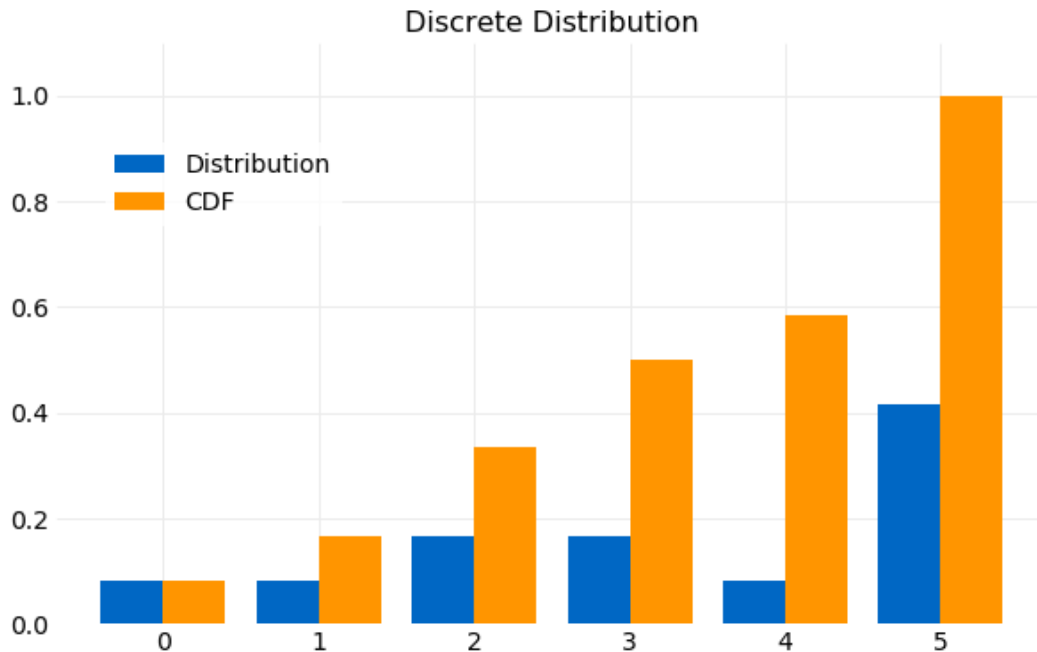
where  $P(N) = 1$ .

For a given generated CDF value,  $u$ , Equation (1) can always be inverted by evaluating it for each  $n$  and searching for the value of  $n$  that satisfies,  $P(n) \geq u$ . It can be seen that the generated samples will have distribution  $\{p_n\}$  since the intervals  $P(n) - P(n-1) = p_n$  are Uniformly sampled.

Consider the distribution,

$$\left\{ \frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{6}, \frac{1}{12}, \frac{5}{12} \right\} \quad (2)$$

It is shown in the following plot with its CDF.



A sampler using the Inverse CDF method can be implemented in Python in a few lines of code,

---

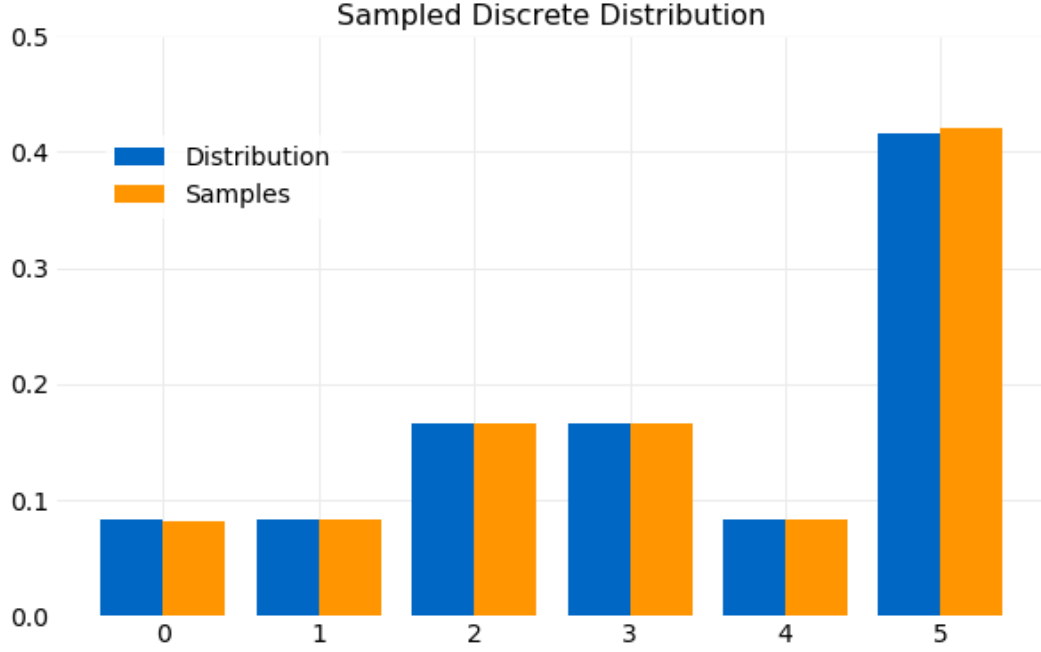
```
import numpy

n = 10000
df = numpy.array([1/12, 1/12, 1/6, 1/6, 1/12, 5/12])
cdf = numpy.cumsum(df)

cdf_star = numpy.random.rand(n)
samples = [numpy.flatnonzero(cdf >= cdf_star[i])[0] for i in range(n)]
```

---

The figure below favoably compares samples generated by the Inverse CDF sampler and distribution (2),



It is also possible to directly sample  $\{p_n\}$  using the `multinomial` sampler from `numpy`,

---

```
import numpy

n = 10000
df = numpy.array([1/12, 1/12, 1/6, 1/6, 1/12, 5/12])
samples = numpy.random.multinomial(n, df, size=1)/n
```

---

### 3 Sampling Continuous Distributions

A continuous probability distribution is defined by the [PDF](#),  $f_X(x)$ , where  $f_X(x) \geq 0, \forall x$  and  $\int f_X(x)dx = 1$ .

The CDF is a monotonically increasing function that specifies probability that  $X \leq x$ ,

$$P(X \leq x) = F_X(x) = \int^x f_X(w)dw \quad (3)$$

To prove that Inverse CDF sampling works is necessary to show that,

$$P[F_X^{-1}(u) \leq x] = F_X(x),$$

where  $F_X^{-1}(x)$  is the inverse of  $F_X(x)$  and  $u$  is Uniform on  $[0, 1]$ . Since  $F_X(x)$  is monotonically increasing,

$$\begin{aligned}
P[F_X^{-1}(u) \leq x] &= P[F_X(F_X^{-1}(u)) \leq F_X(x)] \\
&= P[u \leq F_X(x)] \\
&= \int_0^{F_X(x)} dw \\
&= F_X(x)
\end{aligned}$$

and the desired result is obtained.