五：抽样方法（一）

1.反函数法

令$u_1,...,u_k$,服从均匀分布，让$x_i = F^{-1}(u_i), x\ k\ = F^{-1}(u_i)$，那么$x_i$ 服从分布F。

```
import org.apache.log4j.{Level, Logger}
import org.apache.spark.{SparkConf, SparkContext}
import org.apache.spark.mllib.random.RandomRDDs._

val conf = new SparkConf().setAppName("run").setMaster("local[2]")
val sc = new SparkContext(conf)
val unif = uniformRDD(sc,1000,1).map(x => -math.log(x))
print(unif)
import java.util.concurrent.ThreadLocalRandom
val r = ThreadLocalRandom.current()
print(r)
val OurExpRVs = (1 to 1000).map( x =>r.nextDouble).map(x => -math.log(x)).toArray[Double]
```

2.拒绝接受法

- 找到c与g(x)，使得$f(x) \leq cg(x)$ 在二者的定义域内恒成立
- 产生密度函数是g_y 的随机数y_i ，以及服从均匀分布的随机数u_i
- 如果$u_i \leq \frac{f(x)}{cg(x)}$成立，那么认为此$y_i$ 是来自$f(x)$ 的分布。
- 因为可以证明的是$f(x|U \leq U(y)) = f(x)$
- c越小，接受度越高
- $c_{optimal} = min_\theta max_x \frac{f(x)}{g_\theta(x)}$
- 拉普拉斯分布可以用来产生正态分布随机数

3.重要抽样法（SIR）

- $w(X^{(j)}) = \frac{f(X^{(j)})}{g(X^{(j)})}$,计算每个样本的权重，以及对应概率$w_j = \frac{w(X^{(j)})}{\sum_{i=1}^{100} w(X^{(i)})}$
- 生成i.i.d的备选样本$x_1,....,x_n$ 服从$g(x)$
- 以概率w_j 再从$x_1,....,x_n$ 抽出m个样本
- 此时的$x_i(i=1,..,m)$近似服从$f(x)$
- j/m ，越大越好，一般来说1要大于10或20