Python 随机现象模拟

生成随机数

分布函数计算

Draw nice picture and make animation

随机模型模拟

1 导入随机模块random

import random

该随机模块基于Mersenne Twister随机数生成器，该随机数生成器经过广泛测试，能够生成53bit精度的浮点数并且周期为219937-1。此模块中还包含分布函数，比如均匀分布，三角分布，Beta分布，指数分布，Gamma分布，高斯分布，正则分布，对数正则分布与Weibull分布。

在概率中最核心的概念为random variate，那么使用该模块生成一个随机变量的方式为：

1 import random # 导入模块

2 g=random.Random # 实例化生成器

3 g.seed(1234) #设置随机种子

4 最后可以生成具有不同分布的随机变量了

0~1的随机数 g.random() 均匀分布

(a,b)之间的浮点随机数 g.uniform(a,b) 均匀分布

[a,b]之间的整数随机数 g.randint(a,b)

从某个列表中进行随机抽样 g.sample(population,k) 在列表population中随机获取k个元素。

所谓离散随机变量，在这里就表现为整数随机数randint，区间决定了样本空间。连续随机变量使用浮点数表示，获得一个近似连续的分布。

一般地我们必须获得均匀分布，这样才有可能获取其他分布形式的随机变量。

g.random()的输出类似于一个列表，只要随机种子相同，那么每次都会生成一个相同的列表，每调用一次g.random()就会输出列表中的下一个数据。如果需要不同的随机数列表，两次过程中设置的随机数种子必须不一样。

Python的科学计算包

scipy与numpy

Python画图包

matplotlib

Scipy包含了针对80多种连续和10种离散随机变量分布的随机变量生成器。对于某个分布，一系列的函数可以使用：

rvs Random Variates gengrator

pdf Probability Density Function 对应于连续随机变量

cdf Cumulative Distribution Function

sf Survival Function 1-CDF

ppf Percent Point Function Inverse of CDF

isf Inverse Survival Function Inverse of SF

stats Return mean,variance,skew,kurtosis

moment non-central moments of the distribution

pmf Probability mass function 对应于离散随机变量

在包含的超过80多个分布中，这些分布的参数使用标准的数学形式，其参数表征了分布的location，scale和形状。可以参考模块文档与概率分布方面的参考书来将参数和实际分布的数学形式对应起来。

使用该模块生成连续随机变量的的方式如下

1 import numpy as mp

import scipy as sp #导入numpy和scipy包，并设置别名以简化书写

import scipy.stats #stats -> statistical package统计学包

2 设置随机数种子。Scipy使用Numpy的随机数生成器。Numpy的种子函数使用方法为：np.random.seed(1234)。

3 生成生成器实例

Normal分布，平均值为10，标准差为4

norm1 = sp.stats.norm(loc=10,scale=4)

0~10的均匀分布

unif1 = sp.stats.uniform(loc=0,scale=10)

平均值为1的指数分布

expo1 = sp.status.expon(scale = 1.0)

4 获取随机值

norm1.rvs()

生成离散随机变量的方式

import numpy as np

import scipy as sp

import scipy.stats

class scipy.stats.rv\_discrete(a=0, b=inf, name=None, badvalue=None, moment\_tol=1e-08, values=None, inc=1, longname=None, shapes=None, extradoc=None, seed=None)

离散随机变量分为两种，第一种是自定义的，自己制定样本空间和样本对应的概率

使用values参数，传入一个tuple(xk,pk)，并且sum(pk)=1。

a,b,inc 定义了样本空间，默认为[0,+∞)，inc为1，即正实数空间。对于一些常见的分布，比如指数分布，泊松分布，需要override \_pmf函数实现，比如：

from scipy.stats import rv\_discrete

class possion\_gen(rv\_discrete):

def \_pmf(self,k,mu):

return exp(-mu)\*mu\*\*k/factorial(k)

如果需要shift该分布，那么只需要调用possion\_gen.pnf(x,u,loc)，其等效于调用了possion\_gen(x-loc,mu)

seed 种子，默认为global的np.random.state,也可以指定为np.random.RandomState,或者为一个正整数值。