孙秋实
ID: 10185501402

Homework - #1

Course: 统计与数据分析基础 Term: Fall 2020

Due Date: 29^{th} September, 2020

Problem 1

分别利用随机投点法和平均值法计算下列定积分

$$J = \int_0^1 log_2(1+x)dx$$

请写出精确值和 n = 103 和 n = 104 时的模拟值，并附上程序（可采用 Python 或 R）

Solution: 先计算精确值：

由题意，

$$J = \int_0^1 \frac{ln(1+x)}{ln(2)}dx = \frac{1}{ln(2)}\int_0^1 ln(1+x)dx = \frac{1}{ln(2)}((1+x)ln(1+x)|_0^1 - \int_0^1 \frac{x+1}{x+1}dx) = \frac{2ln(2)-1}{ln(2)}$$

再使用平均值法进行估计：

```python
import numpy as np
from scipy import integrate

def test_function(x):
    answer = np.log2(x+1)
    return answer
def estimate(N):
    randomly_generated_array = np.random.uniform(0, 1, N)
    #PRINT(RANDOMLY_GENERATED_ARRAY)
    answer_mat = test_function(randomly_generated_array)
    #PRINT(ANSWER_MAT)
    sum = 0
    for i in answer_mat:
        sum = sum + i
    estimation = sum * 1.000 / N
    print("Estimation:", estimation)
    print("Error is approximate to:", estimation-0.5573)
    #真实值(2LN(2)-1)/LN(2) APPROX 0.5573

estimate(5000) #选取N=5000 进行试验
```

[98]:
```
import math
```

[99]:
```
import numpy as np
from scipy import integrate
```

[100]:
```
def test_function(x):
    answer = np.log2(x+1)
    return answer
```

[101]:
```
def estimate(N):
    randomly_generated_array = np.random.uniform(0, 1, N)
    #print(randomly_generated_array)
    answer_mat = test_function(randomly_generated_array)
    #print(answer_mat)
    sum = 0
    for i in answer_mat:
        sum = sum + i
    estimation = sum * 1.000 / N
    print("Estimation:", estimation)
    print("Error is approximate to:", estimation-0.5573)
    #真实值(2ln(2)-1)/ln(2) approx 0.5573
```

[102]:
```
estimate(5000)
```

```
Estimation: 0.5578603939544607
Error is approximate to: 0.000560393954460725
```

再使用随机投点法进行估计:

```
# CODING=UTF-8
import numpy as np
import matplotlib.pyplot as plt
list1=[]
list2=[]

plt.figure()
count=0
n=15000 #做15，000次投掷
left,right=0,1 #约束边界条件
lower,upper=0,1

x=np.random.uniform(left,right,n) #调用均匀分布开始制作点列
y=np.random.uniform(lower,upper,n)

for i in range (0,n-1):
    if(y[i]<=np.log2(x[i]+1)):
        #PRINT(NP.LOG2(X[I]+1))
        list1.append(x[i])
        list2.append(y[i])
        count+=1
        #PRINT (COUNT)
        # 真实值(2LN(2)-1)/LN(2) APPROX 0.5573
        print('iteration:'+str(i))
        estimation = count / 1.000 / n
        print(estimation)
        print('error:'+str(abs(estimation-0.5573)))

fig=plt.figure()
axes=fig.add_subplot(1,1,1)
```

```
31    axes.plot(list1,list2,'ro',label = "Monte Carlo Method",color='deepskyblue',markersize=0.6)
32
33    plt.axis('equal') #防止图形在Jupyter-lab中变形
34    plt.show()
```

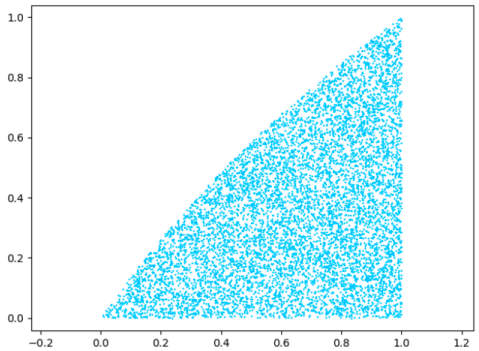

图 1: 可视化随机投点结果

```
Run:  random_toss
iteration:14994
0.555333333333
error:0.00196666666667
iteration:14996
0.5554
error:0.0019
iteration:14997
0.555466666667
error:0.00183333333333
iteration:14998
0.555533333333
error:0.00176666666667
4: Run  6: TODO  Terminal  Python Console  R Console
```