

# 数据挖掘导论作业 1

苏泽华 16340195

## Exercise 1:

代码截图:

```
1 import random
2 import numpy as np
3
4 n = 5000 # 重复次数
5 r = 1.0 # 圆的半径
6 count = 0 # 随机点落在圆内的次数
7 list = [] # 用于存储每轮结果的元组
8
9 for i in range(0,20): # 重复20次
10     for ii in range(0,n): # 每次取样n个
11         x = random.uniform(0,1) # 随机x坐标
12         y = random.uniform(0,1) # 随机y坐标
13         if x*x + y*y <= 1.0: # 根据坐标判断是否在圆内
14             count += 1
15
16     list.append((count/float(n)) * 4) # 计算一次π，存入元组
17     count = 0
18
19 print(np.mean(list)) # 均值
20 print(np.var(list)) # 方差
```

实验结果:

n	均值	方差
50	3.1480	0.068656
100	3.1220	0.050476
200	3.1220	0.135560
300	3.1227	0.010482
500	3.1384	0.006240
1000	3.1250	0.000862
5000	3.1504	0.000444

## Exercise2:

x 可以通过均匀分布、正态分布等多种分布采样获得。但是比较适合的是均匀分布，因为其他分布方式大都具有聚集性，不利于用在区间内通过概率推算积分。

代码截图：

```
1 import random
2 import numpy as np
3
4 n = 100      # 重复次数
5 count = 0    # 随机点落在范围内的次数
6 list = []    # 用于存储每轮结果的元组
7
8 for i in range(0,100): # 重复100次
9     for ii in range(0,n): # 每次取样n个
10        x = random.uniform(0,1) # 随机x坐标
11        y = random.uniform(0,1) # 随机y坐标
12        if x * x * x > y: # 根据随机坐标判断随机点是否在范围内
13            count += 1
14
15    list.append(count/float(n)) # 将得到的积分值放入元组内
16    count = 0
17
18 print(np.mean(list))      # 均值
19 print(np.var(list))      # 方差
```

实验结果：

由下表结果分析可得：当采用均匀分布采样获得 x 时，随着采样次数的增加，实验结果的方差逐渐减小；而随着 n 的增加，均值没有明显的变化态势，均围绕实际值作上下抖动，结果均与实际值较为接近。

均匀分布

n	均值	方差
5	0.25400	0.035884
10	0.27200	0.021216
20	0.25750	0.011369
30	0.25170	0.005564
40	0.25075	0.003930
50	0.24880	0.003499
60	0.24883	0.002574
70	0.24057	0.002394
80	0.24475	0.002019
100	0.25270	0.002039

由下表结果分析可得：当采用正态分布(均值为 0.5，标准差为 0.2)采样获得  $x$  时，随着采样次数的增加，实验结果的方差逐渐减小；而随着  $n$  的增加，其均值围绕 0.12 抖动，但与实际值相差较大。

正态分布

n	均值	方差
5	0.1080	0.017900
10	0.1340	0.014044
20	0.1105	0.004315
30	0.1213	0.003120
40	0.1250	0.002687
50	0.1136	0.002263
60	0.1234	0.001256
70	0.1227	0.001563
80	0.1211	0.001583
100	0.1254	0.000876

### Exercise3:

该积分不宜通过公式直接求解。如果使用蒙特卡洛的方法求解该积分，我认为  $(x,y)$  可以通过均匀分布、正态分布等多种分布采样获得。但是比较适合的是均匀分布，因为其他分布方式大都具有聚集性，不利于用在区间内通过概率推算积分。

代码截图：

```

1  import random
2  import numpy as np
3
4  n = 500      # 采样次数
5  vol = 0      # 体积
6  list = []    # 用于存储每轮结果的元组
7  e = 2.71828 # 自然常数
8
9  for i in range(0,100): # 重复100次
10     for ii in range(0,n): # 每轮采样n个
11         x = random.uniform(2,4) # 随机x坐标
12         y = random.uniform(-1,1) # 随机y坐标
13         xS = pow(x, 2)
14         yS = pow(y, 2)
15         # 大致计算体积(底面积x高)
16         vol += 2*2 * ((yS * pow(e, -yS) + xS*xS * pow(e, -xS)) / (x*pow(e, -xS)))
17
18     list.append(vol/float(n)) # 将得到的积分值放入元组内
19     vol = 0
20
21 print(np.mean(list)) # 均值
22 print(np.var(list)) # 方差

```

### 实验结果：

由下表结果分析可得：当采用均匀分布采样获得(x,y)时，随着采样次数的增加，实验结果的方差逐渐减小，但其方差的总体值依然非常大；而随着 n 的增加，均值没有明显的变化态势，均围绕 110,000 作上下抖动，但由于我们不知道该积分的实际值，无法确认所得均值与实际值之间的误差情况。

n	均值	方差
10	103024.29724173872	12528642941.334663
20	111503.41146260175	5882483775.6046580
30	107000.52044266263	3407950191.7931824
40	108527.27685293477	2894221598.3077345
50	109046.30297622629	2064401979.6569068
60	109694.09751272835	2268150023.6161404
70	107751.02531981321	1565072922.6926942
80	115131.15076138377	1569867965.5832021
100	107630.8732717175	1375305575.9399123
200	115283.96908311741	561778269.93747070
500	113827.55505693323	260233093.34573227