Data Mining Homework 1

学号	姓名	专业	日期
16340311	周远笛	软件工程（数字媒体）	2019.3.6

求pi值

在1*1的区域内，随机坐标点，计算与圆心的距离判断其是否在圆内，并据此根据公式

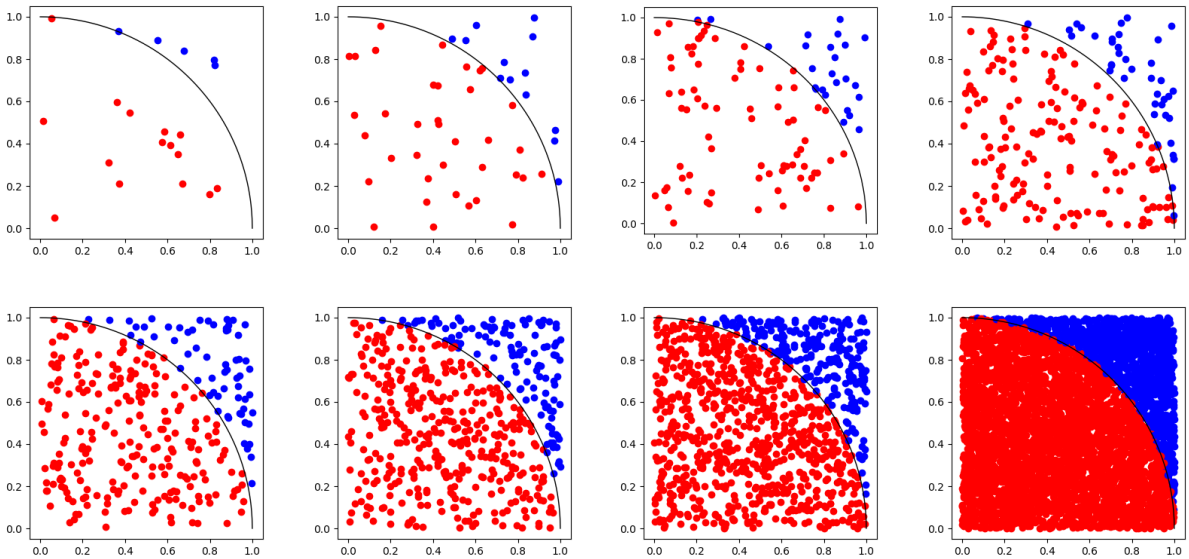
$$S_{扇} = \frac{S_{内}}{N}$$

计算扇形面积.

并根据公式推出pi的计算方法：

$$\pi = S_{扇} * 4$$

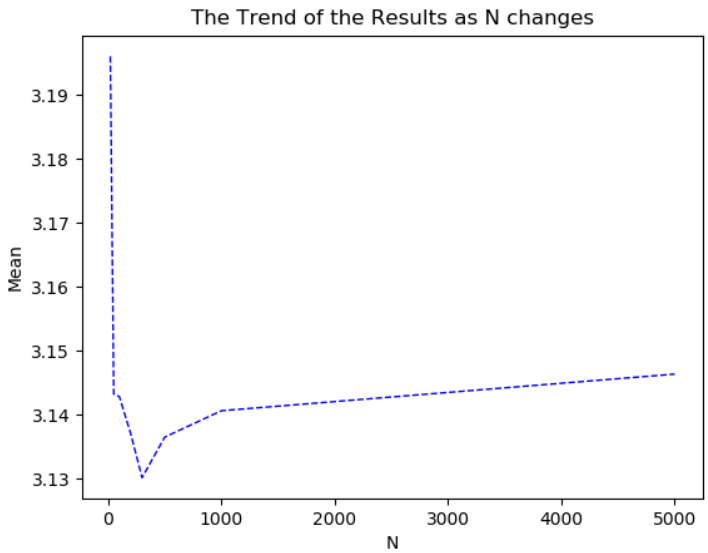
随机采样



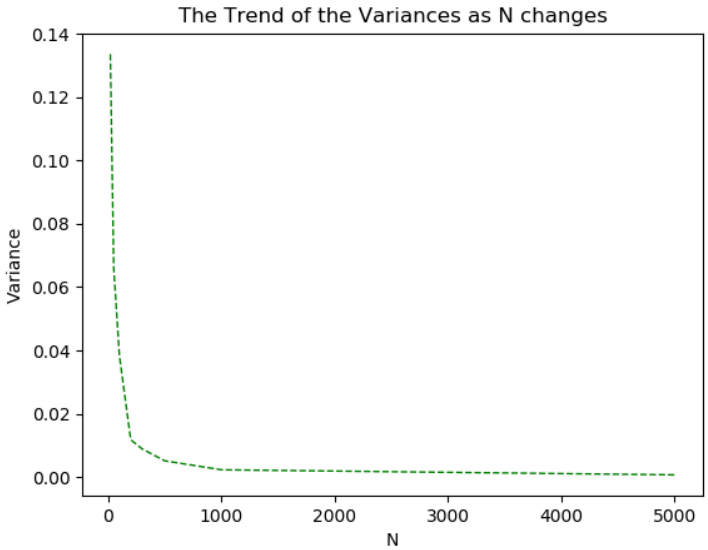
结果

N	mean	variance
20	3.196	0.133584
50	3.1432	0.066086
100	3.1428	0.038376
200	3.137	0.011779
300	3.130133	0.008907
500	3.13648	0.005124
1000	3.1406	0.002293
5000	3.146328	0.000691

可以看出



- 求出的pi在真实值两侧波动。



- 方差随着采样点的增加而降低。

求解简单积分

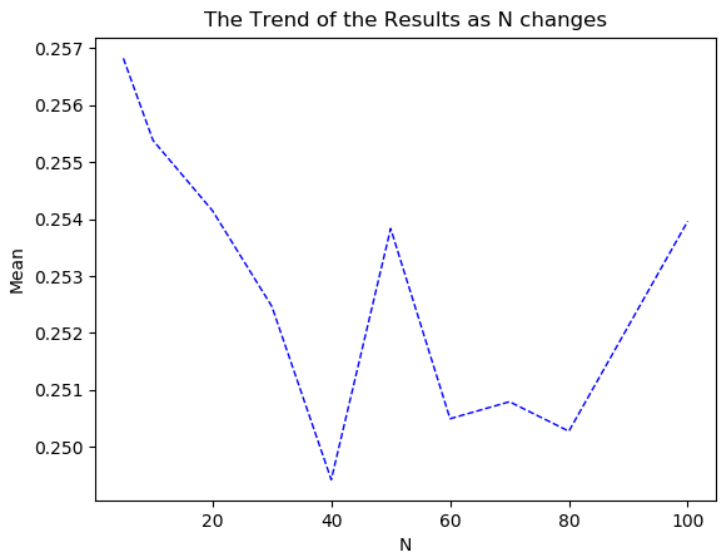
$$\int_0^1 x^3$$

均匀分布采样

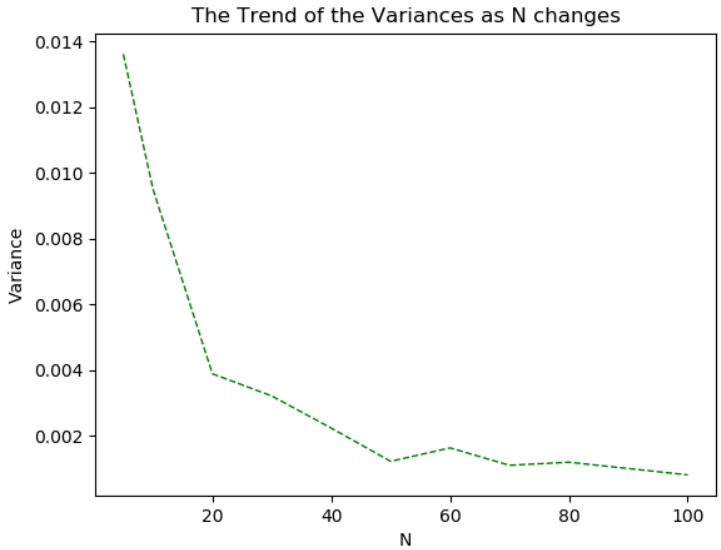
```
x = random.random()
y = pow(x, 3)
```

根据N的值重复上述步骤，每次结果加上`(1-0)*y/N`，计算积分结果的期望，并对此过程重复100次。计算100次内所求的积分结果的均值和方差，结果如下。

N	mean	variance
5	0.256824	0.013612
10	0.255383	0.009499
20	0.25415	0.003882
30	0.252468	0.003206
40	0.249425	0.002229
50	0.253835	0.001218
60	0.250496	0.001627
70	0.250794	0.001097
80	0.250277	0.001191
100	0.253963	0.000806



- 采样取得的结果大部分比真实值0.25大，猜测是均匀分布的随机采样导致的。



- 方差大体上随着N的增加而降低。

求解复杂积分（二重积分）

$$\int_{x=2}^{4}\int_{y=-1}^{1} f(x,y) = \frac{y^2 * e^{-y^2} + x^4 * e^{-x^2}}{x * e^{-x^2}}$$

由于积分过程中涉及到对

$$\int e^{-x^2} dx$$

的求解，因而无法通过公式求解。

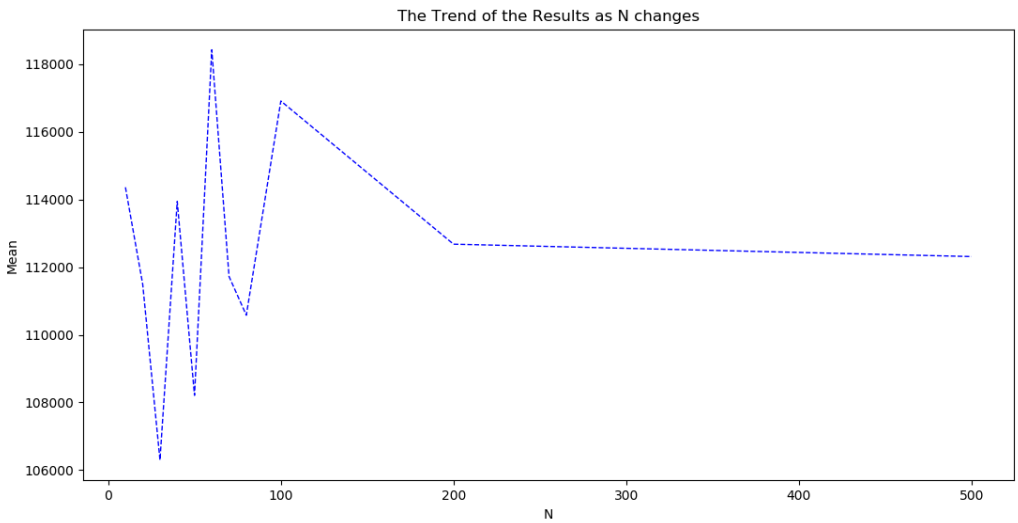
均匀分布采样

假设x, y满足均匀分布，则在范围内每个值的概率均等。直接使用Python `random`包的`random()`实现随机。

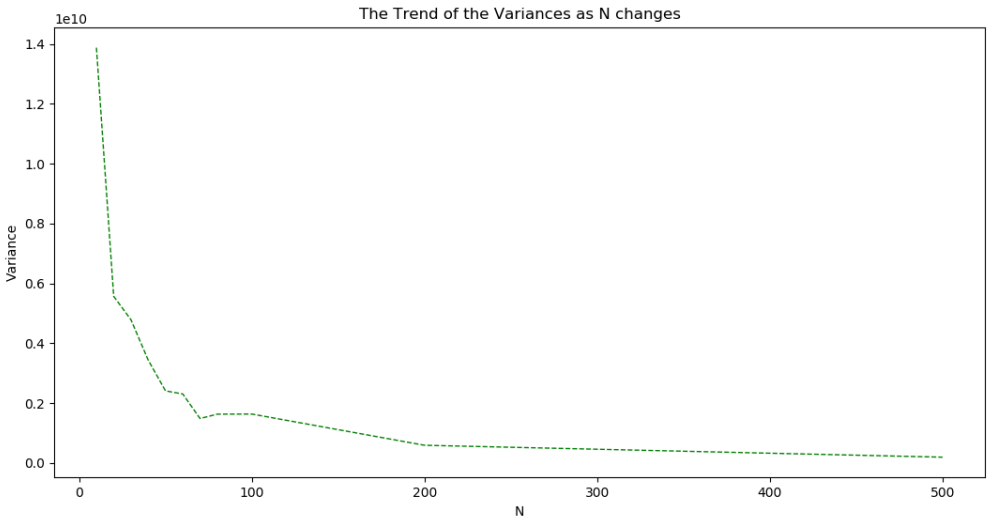
```
x = random.random() * 2 + 2 # x in [2, 4]
y = random.random() * 2 - 1 # y in [-1, 1]
```

重复蒙特卡洛100次后得到的结果如下（输出文件为test3.csv）：

N	mean	variance
10	114360.7	13878564835
20	111494.6	5569440463
30	106297	4787094413
40	113941.8	3436205685
50	108208	2410939505
60	118424	2306120360
70	111722.6	1486991942
80	110579.2	1632578538
100	116911.1	1635650179
200	112678.5	592460179.8
500	112312.4	194789459.6



- 重复100次蒙特卡洛方法可以得到积分结果在正确值两侧振动，且随着重复次数增多，更接近真实结果。



> 纵坐标×pow(10, 10)为真实的方差

- 方差随着N的增大呈下降趋势。

附录：源代码

Question1

```
import random
import numpy as np
import csv
import matplotlib.pyplot as plt


def cal_pi():
    # read user's input
    num_of_points = [20, 50, 100, 200, 300, 500, 1000, 5000]
    # csv version
    csv_file = open("Question1.csv", "w")
    writer = csv.writer(csv_file)
    writer.writerow(['number of points', 'mean', 'variance'])
    # arrays
    mean_array = []
    var_array = []
    # 8 n
    for i in range(8):
        # plt.figure(i, figsize=(4, 4))
        temp = []
        # 20 times
        for t in range(100):
            inside = 0
            for j in range(num_of_points[i]):
                x = random.random()
                y = random.random()
                # inside the quarter of circle
                if x*x+y*y <= 1:
```

```
                    inside = inside + 1
                    # plot the discrete random points in the circle
                    # plt.plot(x, y, 'ro', color="red")
                # else:
                    # plt.plot(x, y, 'ro', color="blue")
            pi = inside * 4 / num_of_points[i]
            temp.append(pi)
        mean_array.append(np.mean(temp))
        var_array.append(np.var(temp))
        row = [num_of_points[i], mean_array[len(mean_array) - 1], var_array[len(var_array)
- 1]]
        writer.writerow(row)
    csv_file.close()
    # calculation finished
    # the following is plotting the figure
    plt.figure(1)
    plt.title("The Trend of the Results as N changes")
    plt.plot(num_of_points, mean_array, "b--", linewidth = 1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Mean")  # Y轴标签
    plt.savefig('Q1_Mean.png')
    plt.figure(2)
    plt.title("The Trend of the Variances as N changes")
    plt.plot(num_of_points, var_array, "g--", linewidth = 1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Variance")  # Y轴标签
    plt.savefig('Q1_Variance.png')


cal_pi()
```

Question2

```
import random
import numpy as np
import csv
import matplotlib.pyplot as plt


def cal_int():
    mean_array = []
    var_array = []
    num_of_points = [5, 10, 20, 30, 40, 50, 60, 70, 80, 100]
    # csv version
    csv_file = open("test2.csv", "w")
    writer = csv.writer(csv_file)
    writer.writerow(['number of points', 'mean', 'variance'])
    # 8 n
    for i in range(len(num_of_points)):
        temp = []
        # 100 times
```

```
        for t in range(100):
            area = 0
            for j in range(num_of_points[i]):
                x = random.random()
                y = pow(x, 3)
                area = area + y
            area = area/num_of_points[i]
            temp.append(area)
        # calculate mean and variance
        mean_array.append(np.mean(temp))
        var_array.append(np.var(temp))
        row = [num_of_points[i], mean_array[len(mean_array) - 1], var_array[len(var_array)
- 1]]
        writer.writerow(row)
    csv_file.close()
    # calculation finished
    # the following is plotting the figure
    plt.figure(1)
    plt.title("The Trend of the Results as N changes")
    plt.plot(num_of_points, mean_array, "b--", linewidth=1)
    # plt.plot(num_of_points, var_array, "r--", linewidth = 1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Mean")  # Y轴标签
    plt.show()
    plt.figure(2)
    plt.title("The Trend of the Variances as N changes")
    plt.plot(num_of_points, var_array, "g--", linewidth=1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Variance")  # Y轴标签
    plt.show()


cal_int()
```

Question3

```
import random
import numpy as np
import csv
import math
import matplotlib.pyplot as plt


def cal_int():
    mean_array = []
    var_array = []
    num_of_points = [10, 20, 30, 40, 50, 60, 70, 80, 100, 200, 500]
    # csv version
    csv_file = open("test3.csv", "w")
    writer = csv.writer(csv_file)
    writer.writerow(['number of points', 'mean', 'variance'])
```

```
    # 8 n
    for i in range(len(num_of_points)):
        temp = []
        # 20 times
        for t in range(100):
            vol = 0
            for j in range(num_of_points[i]):
                x = random.random() * 2 + 2
                y = random.random() * 2 - 1
                z = (pow(y, 2) * math.exp(-pow(y, 2)) + pow(x, 4) * math.exp(-pow(x, 2))) /
(x * math.exp(-pow(x, 2)))
                vol = vol + (4 - 2) * (1 - (-1)) * z
            vol = vol/num_of_points[i]
            temp.append(vol)
        # calculate mean and variance
        mean_array.append(np.mean(temp))
        var_array.append(np.var(temp))
        row = [num_of_points[i], mean_array[len(mean_array) - 1], var_array[len(var_array)
- 1]]
        writer.writerow(row)
    csv_file.close()
    # calculation finished
    # the following is plotting the figure
    plt.figure(1)
    plt.title("The Trend of the Results as N changes")
    plt.plot(num_of_points, mean_array, "b--", linewidth = 1)
    # plt.plot(num_of_points, var_array, "r--", linewidth = 1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Mean")  # Y轴标签
    plt.show()
    plt.figure(2)
    plt.title("The Trend of the Variances as N changes")
    plt.plot(num_of_points, var_array, "g--", linewidth = 1)
    plt.xlabel("N")  # X轴标签
    plt.ylabel("Variance")  # Y轴标签
    plt.show()


cal_int()
```