

In [1]:



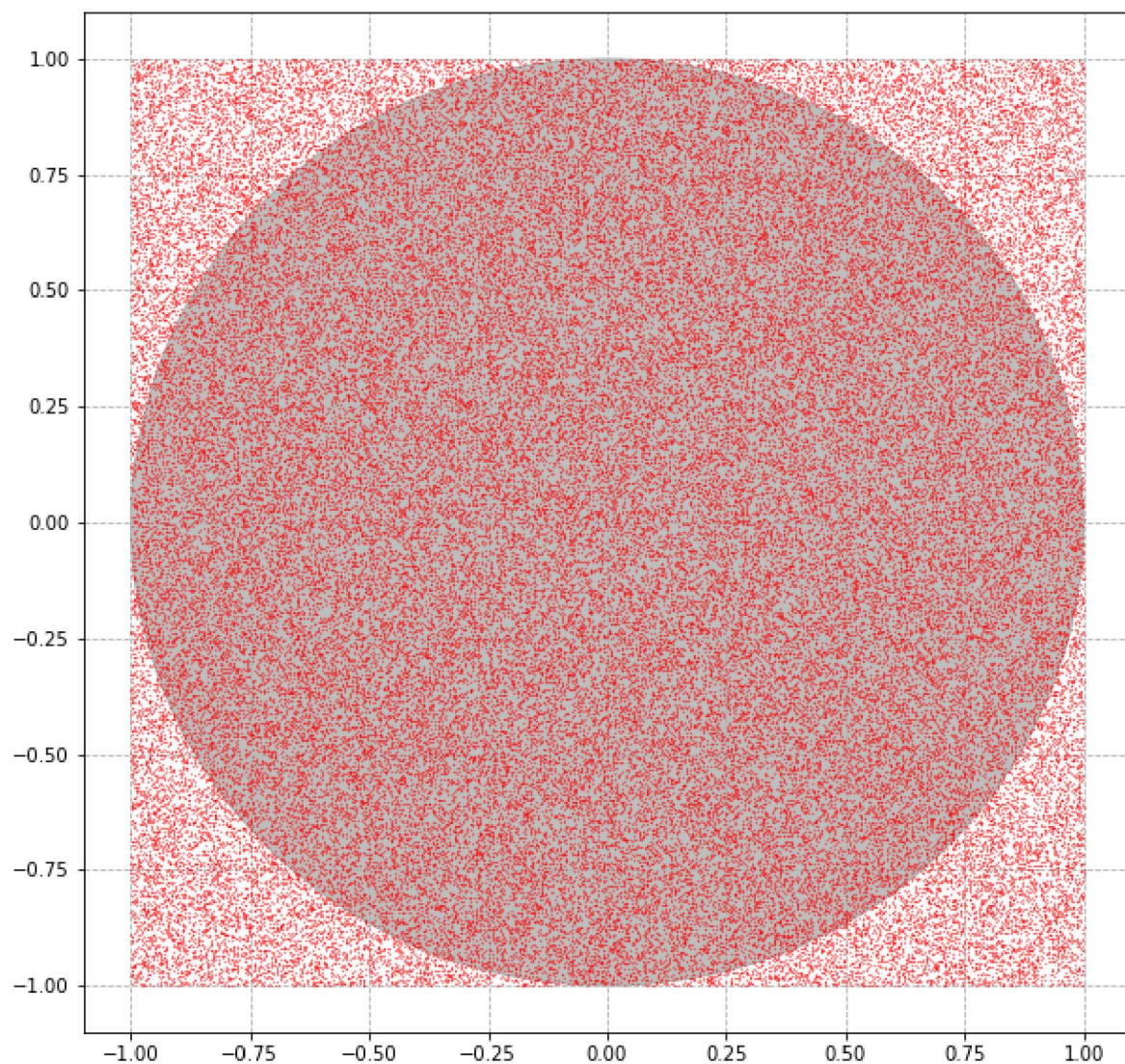
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from matplotlib.patches import Circle
4
5 r = 1 # 半径
6 a, b = (0.0, 0.0) # 圆心
7
8 # 正方形边界
9 xmin, xmax = a-1, a+1
10 ymin, ymax = b-1, b+1
11
12 # 随机抽取n个点
13 n = 100000
14
15 # 在正方形范围内，随机抽取n个点
16 x = np.random.uniform(xmin, xmax, n)
17 y = np.random.uniform(ymin, ymax, n)
18
19 # 计算每个点到圆心的距离
20 d = np.sqrt((x-a)**2 + (y-b)**2)
21
22 # 园内点的个数
23 res = sum(np.where(d<r, 1, 0)) # 距离小于半径即为园内
24
25 # pi
26 pi = res/n*4
27 print(pi)
```

3.14016

In [2]:



```
1 # 绘制正方形
2 fig = plt.figure(figsize=(10, 10))
3 axes = fig.add_subplot(1, 1, 1)
4 plt.plot(x, y, 'ro', markersize=0.2)
5
6
7 # 绘制圆形
8 circle = Circle(xy=(a, b), radius=r, alpha=0.5, color='gray')
9 axes.add_patch(circle)
10 plt.grid(True, linestyle='--', linewidth=0.8)
```



Numpy是python中用来做数学计算常用的工具，import numpy之后就可以使用

Numpy中有很多数学函数和常数，例如

In [3]:



```
1 print(np.pi, np.e)
```

```
3.141592653589793 2.718281828459045
```

In [4]:



```
1 print(np.exp(2.0), np.sin(1.5))
```

```
7.38905609893065 0.9974949866040544
```

Numpy中提供了Ndarray对象用来表示高维数组
用np.array方法可以将列表或元组等对象转换成Ndarray

In [5]:



```
1 np.array((1, 2, 3))
```

Out[5]:

```
array([1, 2, 3])
```

In [6]:



```
1 np.array([[1, 2, 4], [2, 3, 1]])
```

Out[6]:

```
array([[1, 2, 4],
       [2, 3, 1]])
```

Ndarray的维数不限，一维和二维比较常用

用Ndarray来计算可以非常方便地同时修改数组中的每一个数

In [7]:



```
1 a = np.array((1, 2, 3))
2 a + 1
```

Out[7]:

```
array([2, 3, 4])
```

In [8]:



```
1 a * 2
```

Out[8]:

```
array([2, 4, 6])
```

In [9]:



```
1 b = np.array((4, 1, 3))
2 a + b
```

Out[9]:

```
array([5, 3, 6])
```

在二维的情况下可以做一些矩阵操作,例如转置等

In [10]:



```
1 a = np.array((1, 2, 3), (4, 5, 6))
2 print(a)
3 print(a.T)
```

```
[[1 2 3]
 [4 5 6]]
[[1 4]
 [2 5]
 [3 6]]
```