Un po' di conti

$e = 2{,}718281828459\ldots$

```
import numpy as np

def stima_nepero(N):
    sommatoria = 0
    for i in range(0,N):
        somma = 0.0
        addendi = 0
        while somma<1 :
            x = np.random.random_sample()
            somma= somma + x
            addendi = addendi + 1
        sommatoria = sommatoria + addendi
        numero = sommatoria/N
    return(numero)
```

e = 2,718281828459 . . .

```
stima_nepero(10**2)
Out[30]: 2.59

stima_nepero(10**2)
Out[31]: 2.69

stima_nepero(10**2)
Out[32]: 2.66

stima_nepero(10**3)
Out[33]: 2.747

stima_nepero(10**3)
Out[34]: 2.714

stima_nepero(10**3)
Out[35]: 2.726
```

```
stima_nepero(10**4)
Out[36]: 2.7253

stima_nepero(10**4)
Out[37]: 2.7052

stima_nepero(10**4)
Out[38]: 2.7211

stima_nepero(10**5)
Out[39]: 2.71723

stima_nepero(10**5)
Out[40]: 2.72064

stima_nepero(10**5)
Out[41]: 2.71559
```

```
stima_nepero(10**6)
Out[42]: 2.716689

stima_nepero(10**6)
Out[43]: 2.718561

stima_nepero(10**6)
Out[44]: 2.71879

stima_nepero(10**7)
Out[45]: 2.7183193

stima_nepero(10**7)
Out[46]: 2.7186078

stima_nepero(10**7)
Out[48]: 2.7183338
```

$\pi = 3,14159265358979323846264338327950288\ldots$

```
import numpy as np

def stima_pi(N):
    rand = np.random.rand(N)/2
    frand = 3/((1 -rand**2)**0.5)
    return np.sum(frand)/N
```

$\pi = 3,14159265358979323846264338327950288\ldots$

```
stima_pi(10)
Out[50]: 3.181715957967824 7

stima_pi(10)
Out[51]: 3.169756270545920 4

stima_pi(10)
Out[52]: 3.201855346234268

stima_pi(10**2)
Out[53]: 3.145170227382451

stima_pi(10**2)
Out[54]: 3.151336161942787 8

stima_pi(10**2)
Out[55]: 3.131542583745878
```

```
stima_pi(10**3)
Out[56]: 3.141841030693593 8

stima_pi(10**3)
Out[57]: 3.143555621366298 5

stima_pi(10**3)
Out[58]: 3.133526196612647 3

stima_pi(10**4)
Out[59]: 3.139605972343868 5

stima_pi(10**4)
Out[60]: 3.140410142910572

stima_pi(10**4)
Out[61]: 3.139101934994702 7
```

$\pi = 3,14159265358979323846264338327950288\ldots$

```
stima_pi(10**5)
Out[62]: 3.141523743136084

stima_pi(10**5)
Out[63]: 3.140946135603006

stima_pi(10**5)
Out[64]: 3.141636045303459

stima_pi(10**6)
Out[65]: 3.1413480791302275

stima_pi(10**6)
Out[66]: 3.141814724872082

stima_pi(10**6)
Out[67]: 3.1416419695748283
```

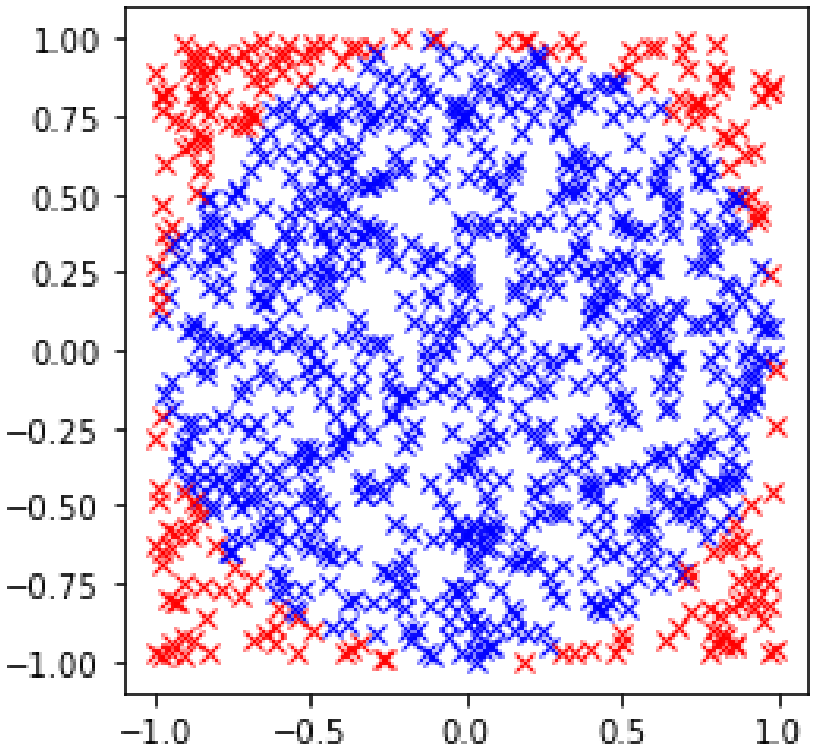
```
stima_pi(10**7)
Out[68]: 3.141551529136709 3

stima_pi(10**7)
Out[69]: 3.141576449575698

stima_pi(10**7)
Out[70]: 3.141559462242652 6

stima_pi(10**8)
Out[71]: 3.141573118798706 6

stima_pi(10**8)
Out[72]: 3.141574821806573 8

stima_pi(10**8)
Out[73]: 3.141580058398877 6
```

$\pi = 3,14159265358979323846264338327950288\ldots$

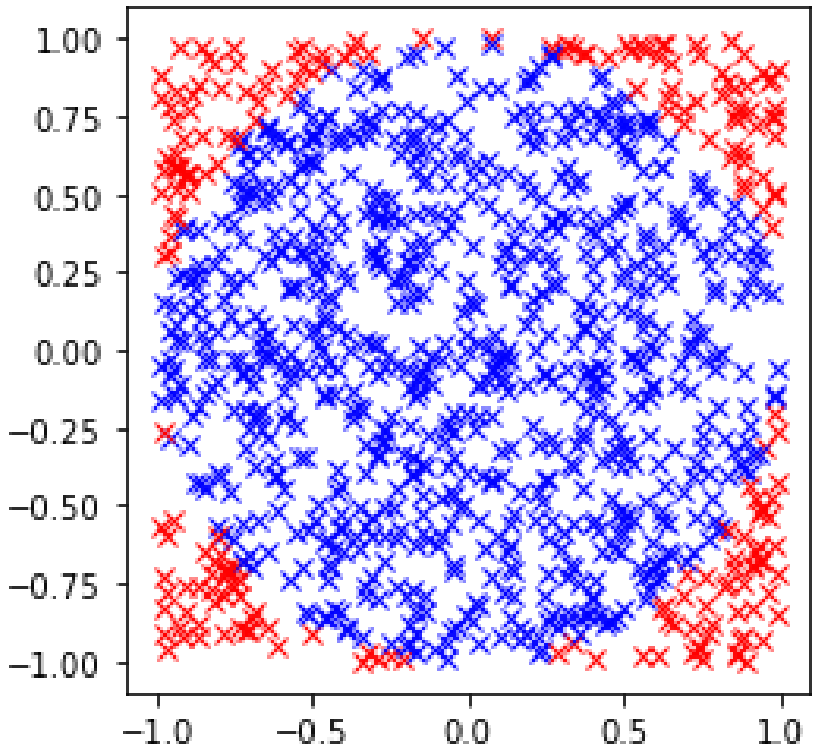
```
import numpy as np
import matplotlib.pyplot as plt

def estimate_pi(N):
    plt.xlim(-1.1,1.1)
    plt.ylim(-1.1, 1.1)
    plt.axes().set_aspect('equal', adjustable='box')
    rand = 2.0*np.random.rand(2, N) - 1.0
    dist = rand[0,:]*rand[0,:] + rand[1,:]*rand[1,:]
    X = rand[0,:]
    Y = rand[1,:]
    for i in range (0, N):
        if X[i]**2 + Y[i]**2 < 1 :
            plt.plot(X[i],Y[i], 'xb')
        else :
            plt.plot(X[i],Y[i], 'xr')
    return np.sum(dist <= 1)*4.0/N
```

$\pi = 3,14159265358979323846264338327950288\ldots$

```
estimate_pi(10**3)
Out[83]: 3.164
```

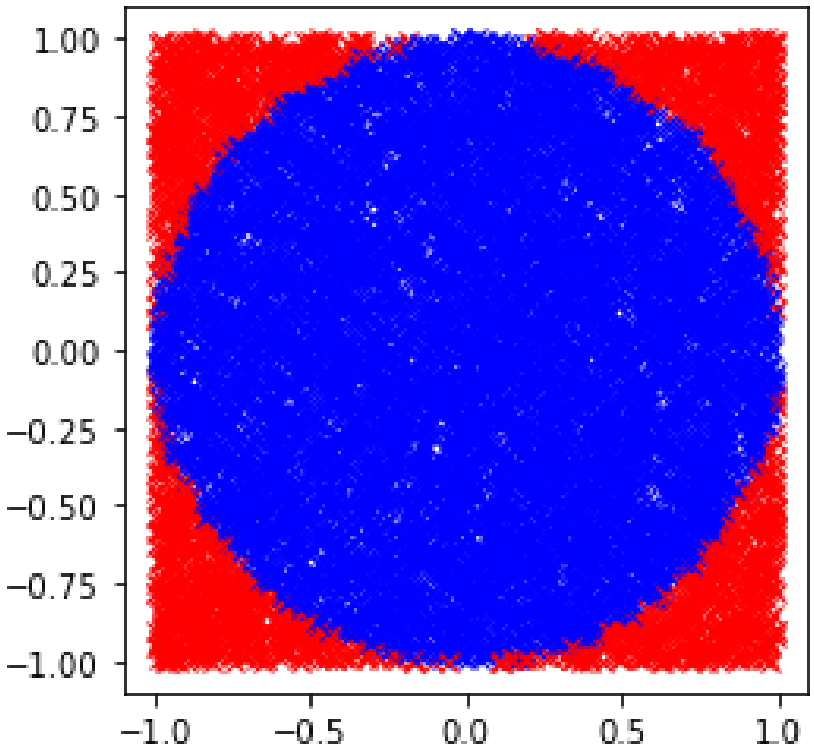



```
estimate_pi(10**3)
Out[82]: 3.128
```

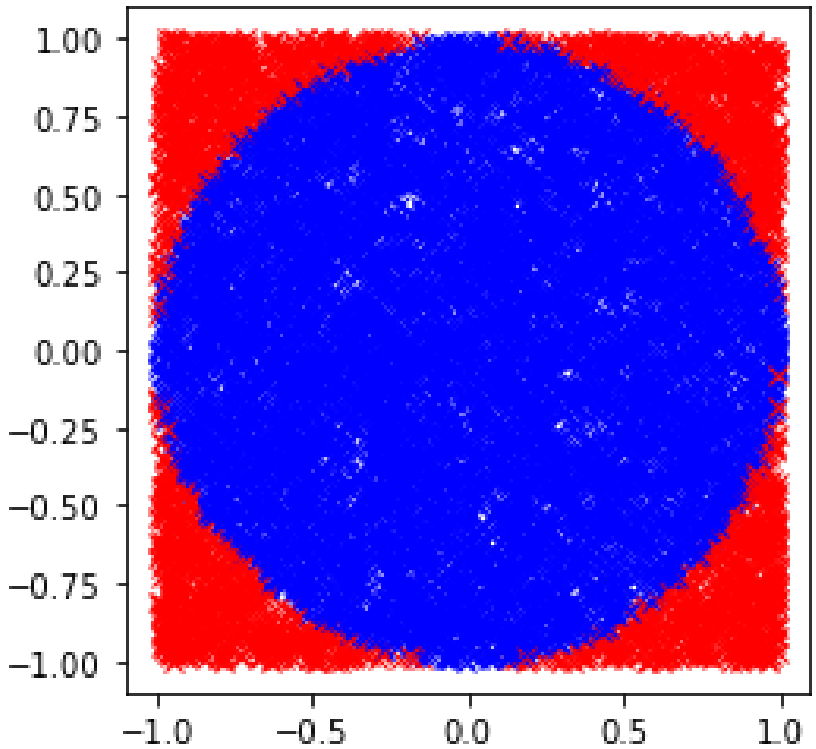


$\pi = 3,14159265358979323846264338327950288\ldots$

```
estimate_pi(10**4)
Out[84]: 3.1336
```




```
estimate_pi(10**4)
Out[86]: 3.1176
```



$\pi = 3,14159265358979323846264338327950288\ldots$

```
estimate_pi(10**5)
Out[90]: 3.14288

estimate_pi(10**5)
Out[91]: 3.1468

estimate_pi(10**5)
Out[92]: 3.13672

estimate_pi(10**6)
Out[93]: 3.144276

estimate_pi(10**6)
Out[94]: 3.138212

estimate_pi(10**6)
Out[95]: 3.14284
```

```
estimate_pi(10**7)
Out[96]: 3.1427456

estimate_pi(10**7)
Out[97]: 3.1420168

estimate_pi(10**7)
Out[98]: 3.1417504

estimate_pi(10**8)
Out[99]: 3.14164384

estimate_pi(10**8)
Out[100]: 3.141396

estimate_pi(10**8)
Out[101]: 3.14153944
```