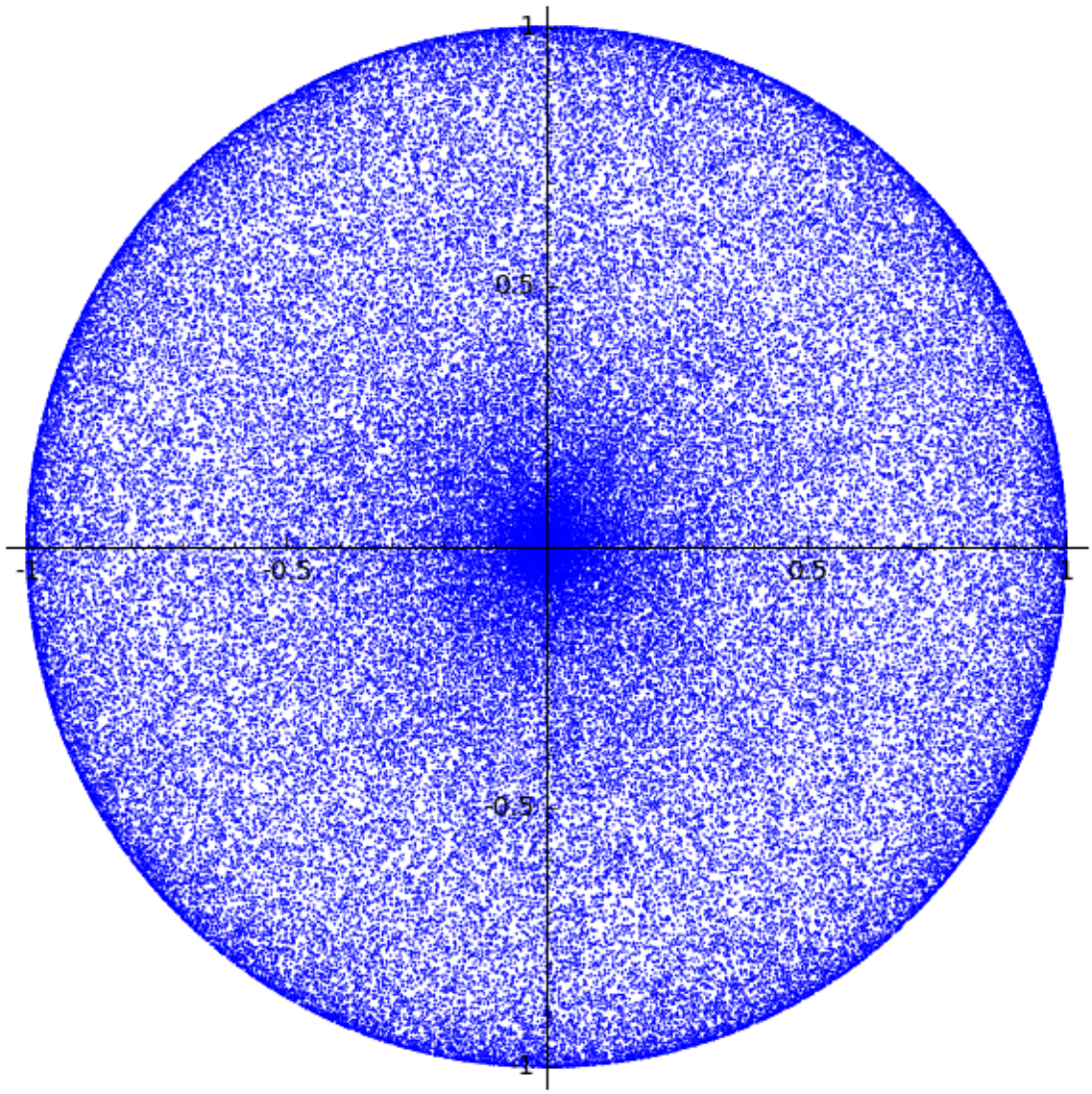
119-PROBA-ejercicios-2-SOL

April 23, 2018

Cuerdas-Ej-11

Usando un poco de trigonometría se puede ver que la longitud del lado del triángulo equilátero inscrito en la circunferencia de radio 1 es $\sqrt{3}$, y también que las cuerdas que miden más de $\sqrt{3}$ son exactamente las que su punto medio dista del origen menos que $1/2$.

```
In [1]: def punto():
            t = 2*pi.n()*random()
            P = cos(t),sin(t)
            return P

In [2]: punto()

Out[2]: (0.0265331962913792, 0.999647932771615)

In [3]: def punto_medio():
            P,Q = punto(),punto()
            return (P[0]+Q[0])/2,(P[1]+Q[1])/2

In [4]: punto_medio()

Out[4]: (-0.0547337893294020, -0.0251181280511781)

In [5]: def prob_cuerda1(R):
            cont = 0
            for muda in xsrange(R):
                P = punto_medio()
                if P[0]^2+P[1]^2 < 1/4:
                    cont += 1
            return (cont/R).n()

In [6]: prob_cuerda1(10^5)

Out[6]: 0.334940000000000

In [7]: def grafica1(R):
            points([punto_medio() for muda in xsrange(R)],pointsize=1).show(aspect_ratio=1)

In [8]: grafica1(10^5)
```



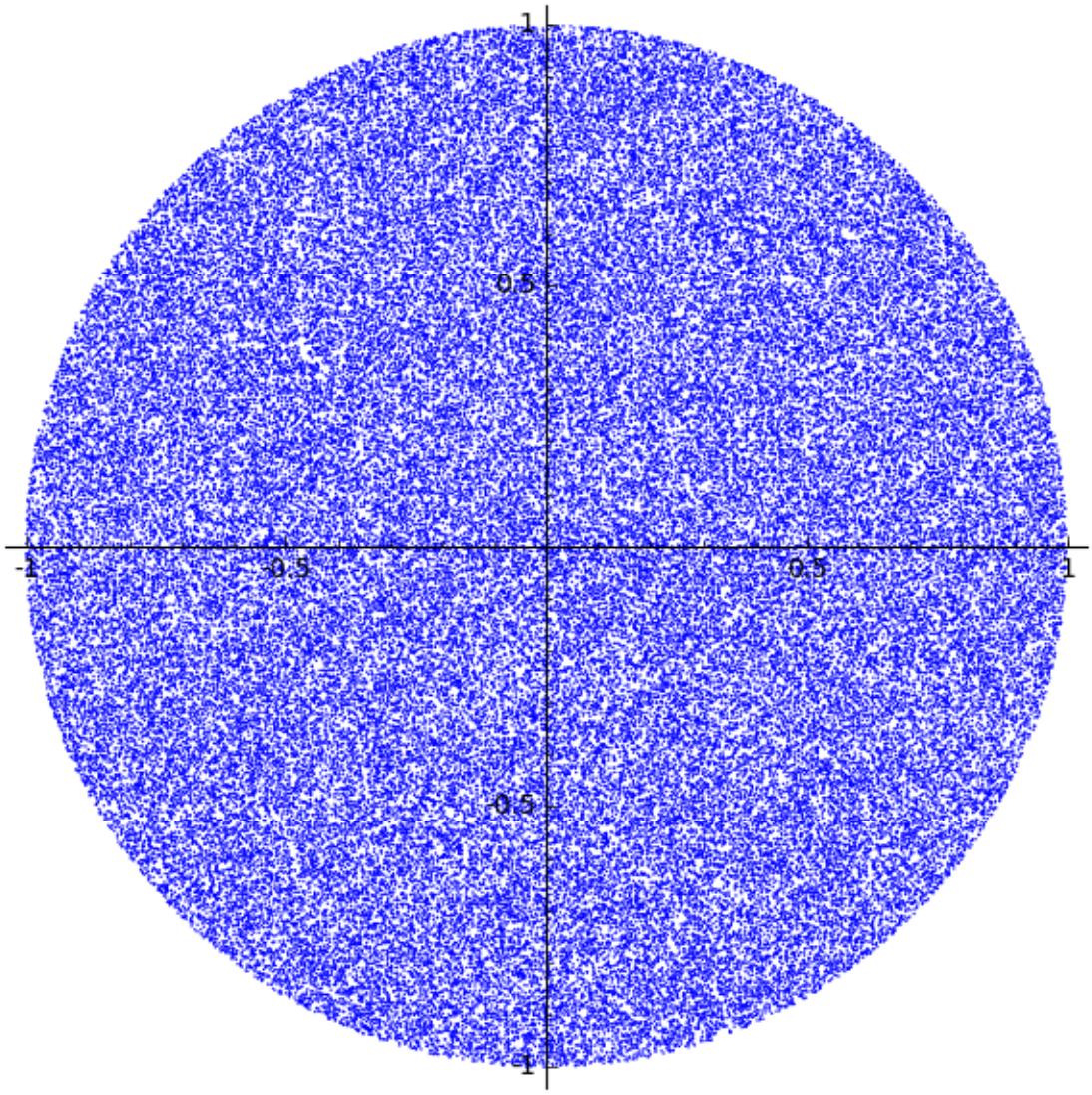

```
In [9]: def prob_cuerda2(R):
            cont = 0
            cont2 =0
            while cont2 < R:
                x,y = 2*random()-1,2*random()-1
                if x^2+y^2 <= 1:
                    cont2 += 1
                if x^2+y^2 < 1/4:
                    cont += 1
            return (cont/R).n()
```

```
In [10]: prob_cuerda2(10^5)
```

```
Out[10]: 0.247080000000000
```

```
In [11]: def grafica2(R):
             cont2 =0
             L = []
             while cont2 < R:
                 x,y = 2*random()-1,2*random()-1
                 if x^2+y^2 <= 1:
                     cont2 += 1
                     L.append((x,y))
             return points(L,pointsize=1).show(aspect_ratio=1)
```

```
In [12]: grafica2(10^5)
```




```
In [13]: def prob_cuerda3(R):
             cont = 0
```

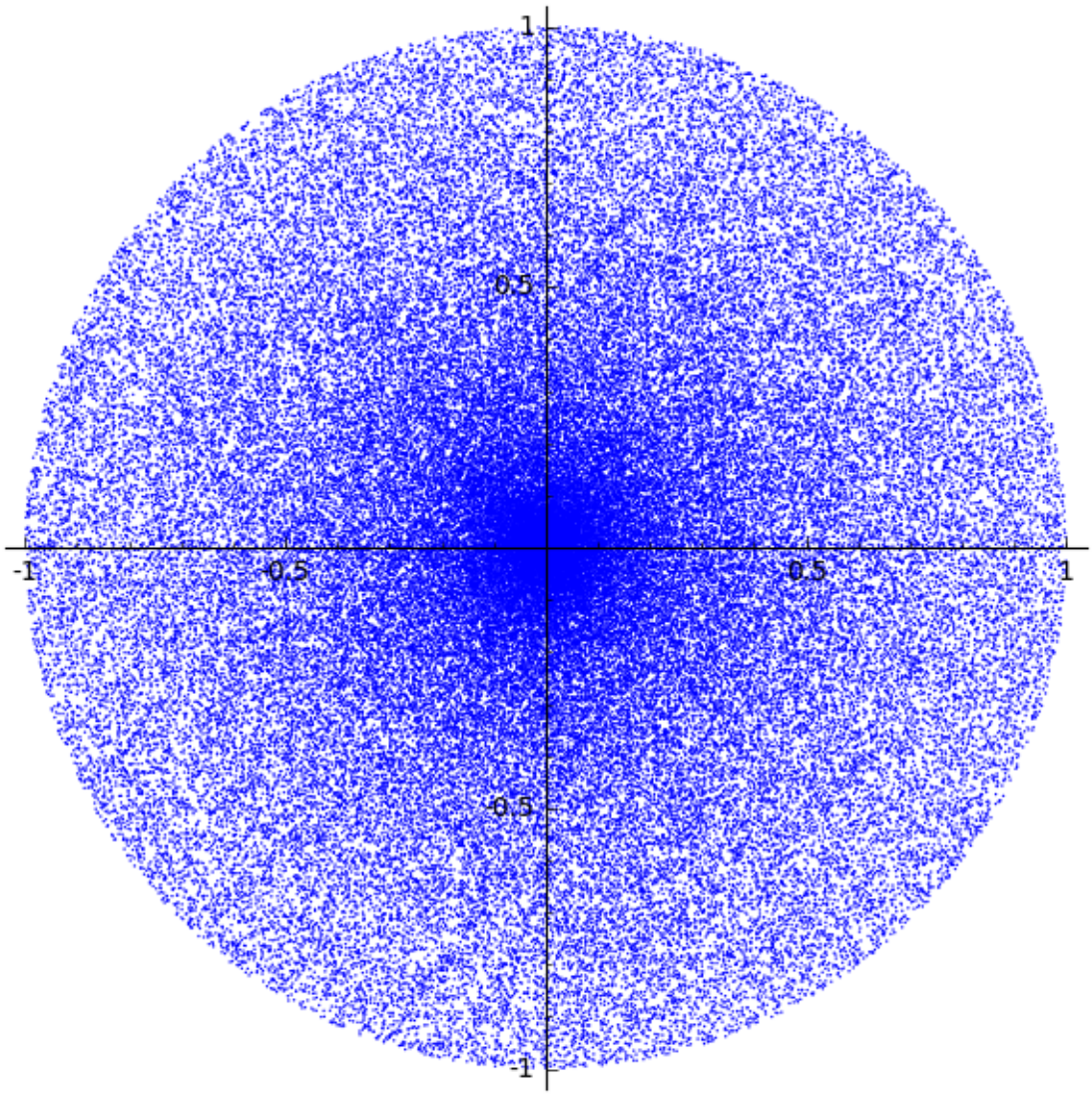
```
            for muda in xsrange(R):
                P = punto()
                r = random()
                Q = (r*P[0],r*P[1])
                if Q[0]^2+Q[1]^2 < 1/4:
                        cont += 1
            return (cont/R).n()
```

```
In [14]: prob_cuerda3(10^5)
```

```
Out[14]: 0.499250000000000
```

```
In [15]: def figura3(R):
            L = []
            for muda in xsrange(R):
                P = punto()
                r = random()
                Q = (r*P[0],r*P[1])
                L.append(Q)
            return points(L,pointsize=1).show(aspect_ratio=1)
```

```
In [16]: figura3(10^5)
```



Cumpleaños-Ej-5.3

```
In [17]: def prob_cumple(N,R):
             cont = 0
             for muda in xsrange(R):
                 A = set([randint(1,365) for muda in srange(N)])
                 if len(A) < N:
                     cont += 1
             return (cont/R).n()

In [18]: [prob_cumple(10*k,10^4) for k in srange(1,7)]

Out[18]: [0.114600000000000,
          0.414900000000000,
```

```
0.711200000000000,
0.889300000000000,
0.967700000000000,
0.994800000000000]
```

Sombreros-Ej-5.2

```
In [19]: def prob_sombreros(N,R):
             cont = 0
             for muda in xsrange(R):
                 for k in srange(N):
                     x = randint(1,N)
                     if x == k:
                         cont += 1
                         break
             return (cont/R).n()

In [20]: prob_sombreros(20,10^5)

Out[20]: 0.624160000000000

In [21]: prob_sombreros(30,10^5)

Out[21]: 0.625150000000000

In [22]: 1-(1/e).n()

Out[22]: 0.632120558828558
```