In [1]:

```
pi
```

Out[1]:

```
π = 3.1415926535897...
```

Area = πr^2

Circ = $\pi 2r$

$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$

$\pi = \sqrt{6 \sum_{k=1}^{\infty} \frac{1}{k^2}}$

In [5]:

```
sqrt(6*(1/1^2 + 1/2^2 + 1/3^2 + 1/4^2 + 1/5^2 + 1/6^2 + 1/7^2))
```

Out[5]:

```
3.011773947846214
```

In [12]:

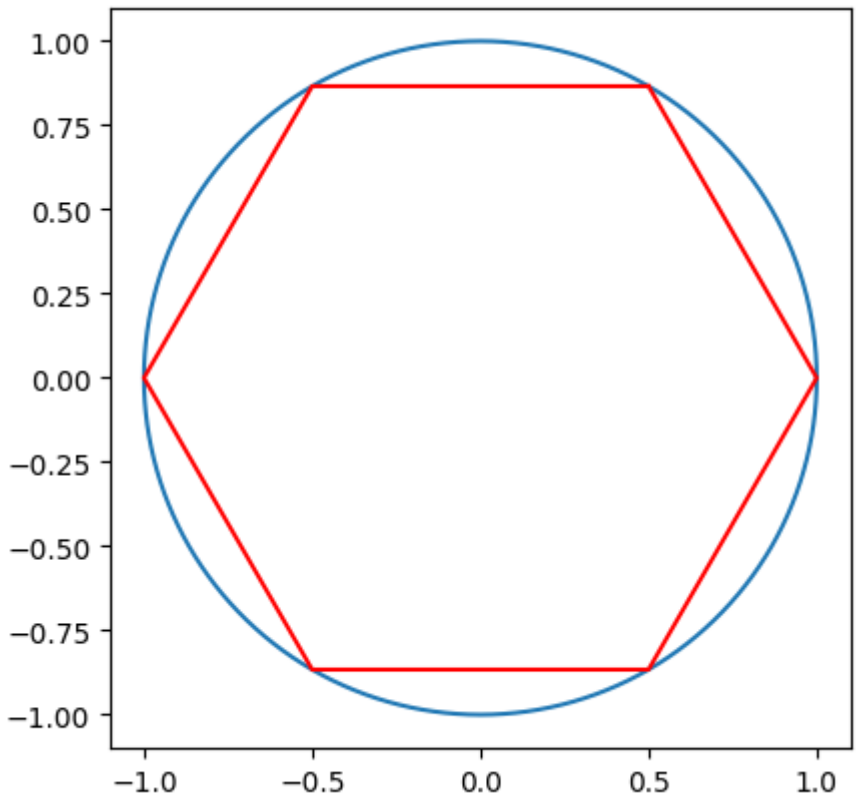
```
n = 10000
sqrt( 6*sum([1/k^2 for k in 1:n]) )
```

Out[12]:

```
3.1414971639472102
```

In [15]:

```
using PyPlot
circlePts = [ [cos(deg2rad(a)),sin(deg2rad(a))] for a in 0:1:360]

subplot(111,aspect=1)
plot(first.(circlePts),last.(circlePts))

function makePts(n)
    th = 360/n
    return [ [cos(deg2rad(k*th)),sin(deg2rad(k*th))]   for k in 1:n+1]
end

pts = makePts(6)
plot(first.(pts),last.(pts),"r")

function dist(w,v)
   sqrt((w[1]-v[1])^2 + (w[2]-v[2])^2)
end

sum([ dist(pts[k],pts[k+1]) for k in 1:length(pts)-1])/2
```

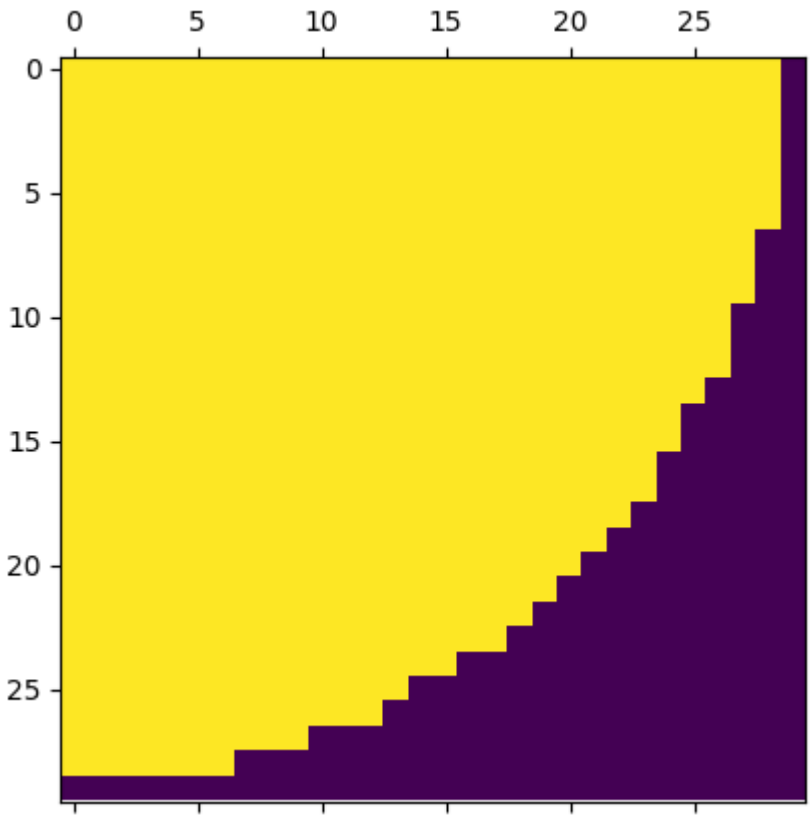


Out[15]:

```
3.0
```

In [17]:

```
n = 30
mat = zeros(n,n)

for i in 1:n
   for j in 1:n
        if i^2 + j^2 <= n^2
           mat[i,j] = 1
        end
    end
end

matshow(mat)

4*sum(mat)/n^2
```

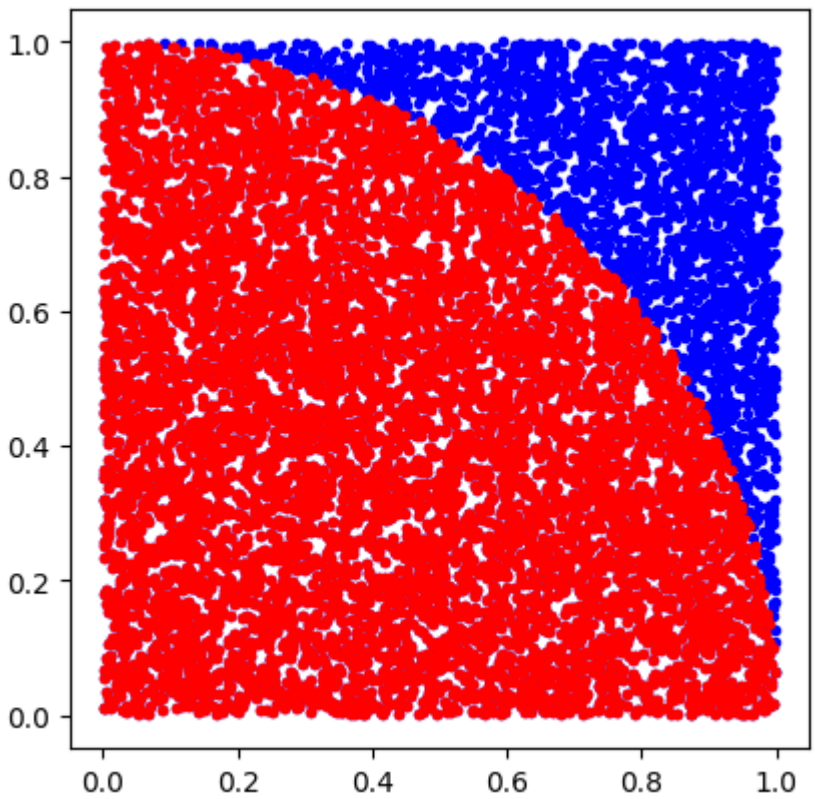


Out[17]:

```
3.0
```

In [102]:

```
n = 10^4
pts = [ [rand(), rand()] for _ in 1:n]
subplot(111,aspect=1)
plot(first.(pts),last.(pts),"b.")

inCirc(pt) = pt[1]^2 + pt[2]^2 <= 1
circPts = filter(inCirc,pts)
plot(first.(circPts),last.(circPts),"r.")
4*length(circPts)/n
```



Out[102]:

```
3.1408
```

In [93]:

```
length(pts),length(circPts)
```

Out[93]:

```
(1000, 795)
```

In [87]:

```
? filter
```

search: **filter filter! fieldtype fill_between fill_betweenx**

Out[87]:

```
  filter(f, a::AbstractArray)

  Return a copy of a, removing elements for which f is false. The func
tion f
  is passed one argument.

  Examples
  ===========

  julia> a = 1:10
  1:10

  julia> filter(isodd, a)
  5-element Array{Int64,1}:
   1
   3
   5
   7
   9

  ____________________________________________________________________
________

  filter(f, d::AbstractDict)

  Return a copy of d, removing elements for which f is false. The func
tion f
  is passed key=>value pairs.

  Examples
  ===========

  julia> d = Dict(1=>"a", 2=>"b")
  Dict{Int64,String} with 2 entries:
    2 => "b"
    1 => "a"

  julia> filter(p->isodd(p.first), d)
  Dict{Int64,String} with 1 entry:
    1 => "a"
```

In [ ]:

```

```