QAMT workshop, February 2019

This notebook contains most of key elements from Session 3 of the MATH TEACHERS CODE Series, 2018.

Follow us on twitter: [@OneOnEpsilon (https://twitter.com/oneonepsilon)](https://twitter.com/oneonepsilon)

Computing Pi via a series approximation

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

In [1]:

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

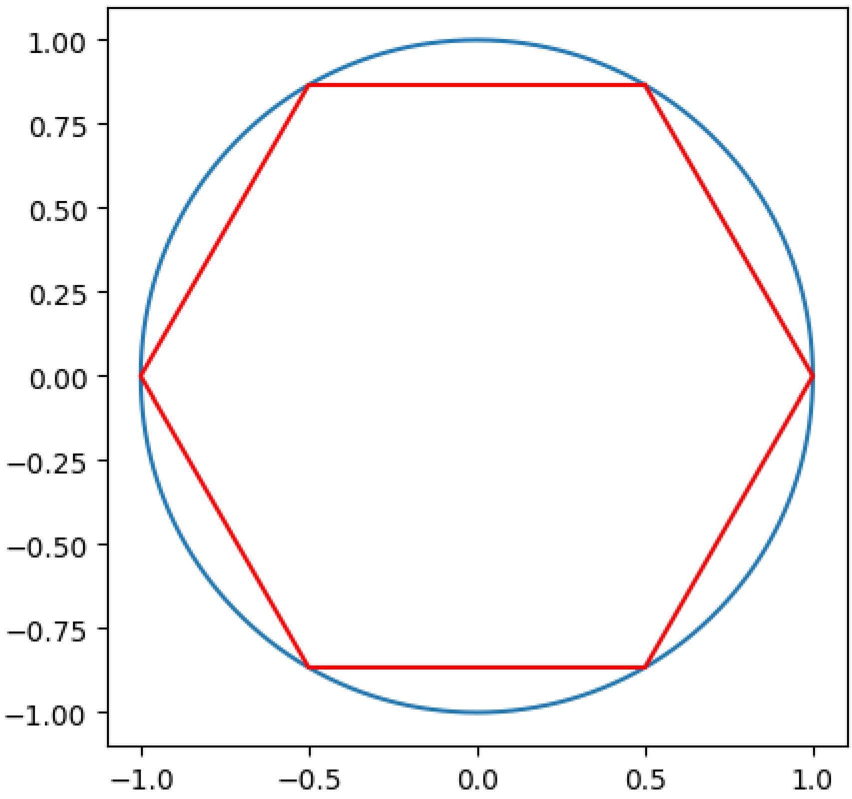
Out[1]:

```
3.1414971639472102
```

Computing Pi by inscribing regular polygons

In [2]:

```
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
```



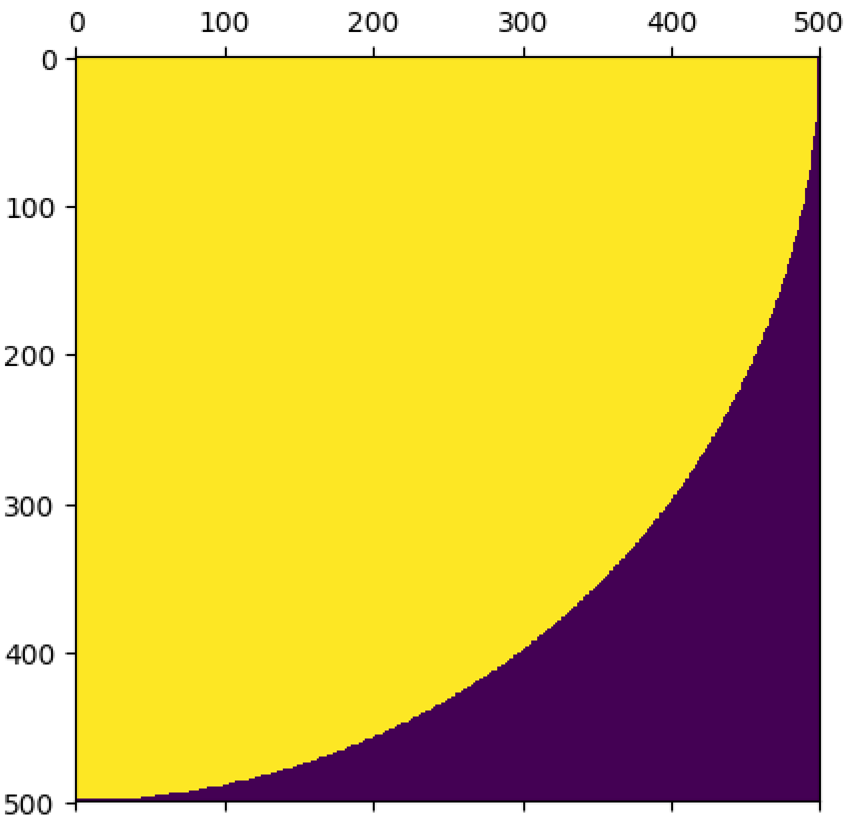
Out[2]:

```
3.0
```

Computing Pi by counting pixels

In [3]:

```julia
using PyPlot

n = 500
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
```



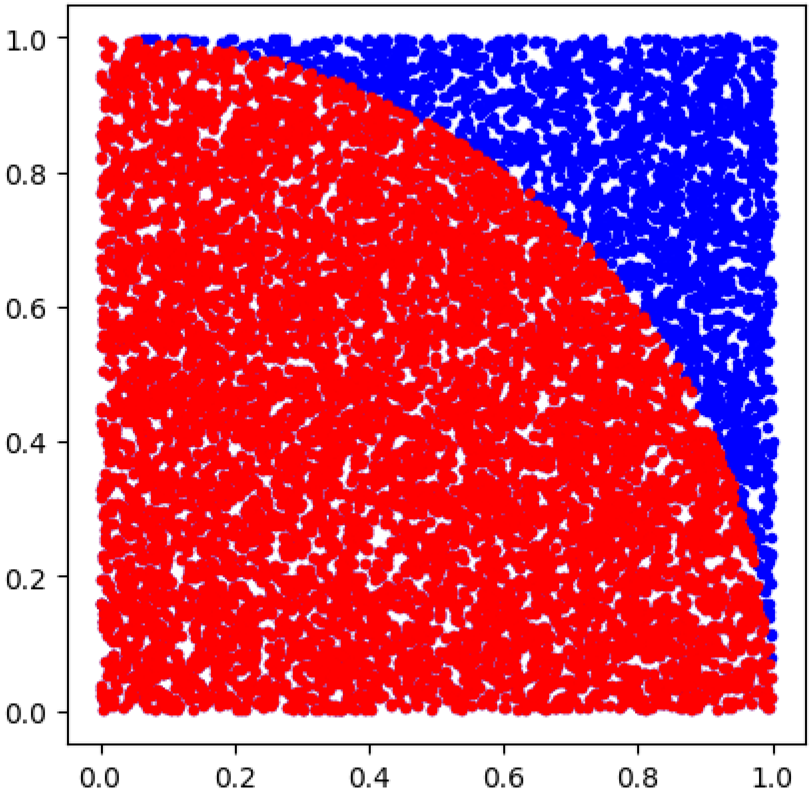
Out[3]:

```
3.133392
```

Computing Pi with random points

In [4]:

```
using PyPlot
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[4]:

```
3.1496
```