PS7

1. We would like to you play with the famous quarter circle law which states that if you draw a scaled histogram of the eigenvalues of a random matrix $\frac{1}{\sqrt{n}}AA^T$, where A is normally distributed, the picture if scaled correctly looks like a quarter circle. (This is exactly the case as $n \to \infty$ but for much smaller n, the quarter circle law works pretty well too.)

The scaled histogram is usually drawn so that the area under the bars is exactly 1. This is known as a probability density. The area under the curve from a to b is then the probability that an eigenvalue chosen uniformly at random is between a and b.

Most languages have a scaled option for histogram plots though MATLAB may be the only one that does not offer that out of the box, though googling MATLAB SCALE HISTOGRAM seems to provide simple approaches. (or simply if you divide the counts by (n * binwidth) the area will be 1)

In Julia you can see a very nice illustration of the quarter circle law by typing

In[1]:
```
using PyPlot

n=2000
a=randn(n,n)
ev=sqrt(eigvals(transpose(a)*a))/sqrt(n);

f=figure(figsize=(5,5))
plt.hist(ev, bins=40, normed="true",color="y")
x=0:.01:2;
plot(x,sqrt(4-x.^2)/pi,"r",linewidth=3)
```

Give this a try and enclose a plot. Use the known integral

$$\int_{-2}^{x} \frac{1}{\pi}\sqrt{4-t^2}dt = \frac{1}{2\pi}(x\sqrt{4-x^2} + 4\arcsin(x/2)),$$

to figure out the probability that an eigenvalue of $A^T A/\sqrt{n}$ is between 0 and 1.

This is a special case of the Marcenko-Pastur law which has many applications in finance.

2. Take three 2 x 2 matrices and compute the image of the unit circle . In other words for all $\|x\| = 1$ plot Ax. One convenient way is to plot a point in the xy plane corresponding to $A * (\cos t, \sin t)^T$, for $0 \leq t \leq 2\pi$.

In Julia

In [2]:
```
function ellipse(A)
for t=[0:.1:2pi]
   plot(cos(t),sin(t),"b.")
   plot((A*[cos(t);sin(t)])... ,"r.")
end
axis("equal")
end
```

Example

In [3]:
```
using PyPlot
ellipse([6 2; 2 1])
```