In [1]:

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

In [35]:

```
nls = [5, 10, 25, 50, 100]
#NPOINTS is for expected number of roots, using WLLN
NPOINTS = 5000
```

In [70]:

```
def poly_root(string, name):
    expected_real = np.zeros(len(nls))
    for idx, n in enumerate(nls):
        plt.figure(idx + 1)
        for itera in range(NPOINTS):
            coeff = np.zeros(n)
            coeff = string(n)
            root = np.roots(coeff)
            expected_real[idx] += np.isreal(root).sum()
        expected_real[idx] /= NPOINTS

        #plotting roots in complex plane
        plt.plot(root.real, root.imag, 'o', label = 'roots')
        plt.xlabel('Real Part')
        plt.ylabel('Imaginary Part')
        plt.legend()
        plt.title('Roots in the Complex Plane for  ' + name + ' coeffecients ' + st
        plt.savefig('Image-Q6/' + name + '-' + str(n) +'.png')

    #plotting n vs expected number of real roots
    plt.figure(len(nls) + 1)
    plt.plot(nls, expected_real, 'go--')
    plt.xlabel('Degree of Polynomial')
    plt.ylabel('Expected Number of Real Roots')
    plt.title('Expected Number of Real Roots for ' + name + ' coeffecients ' + str(
    plt.savefig('Image-Q6/' + name + '-real-roots-' + str(n) +'.png')
    return expected_real
```

binomial distribution

In [53]:

```
def binom(n):
    x = np.random.uniform(low = 0.0, high=1.0, size = n)
    y= np.zeros(n)
    idx = x >0.5
    y[idx] = 1
    y[~idx] = -1
    return y
```

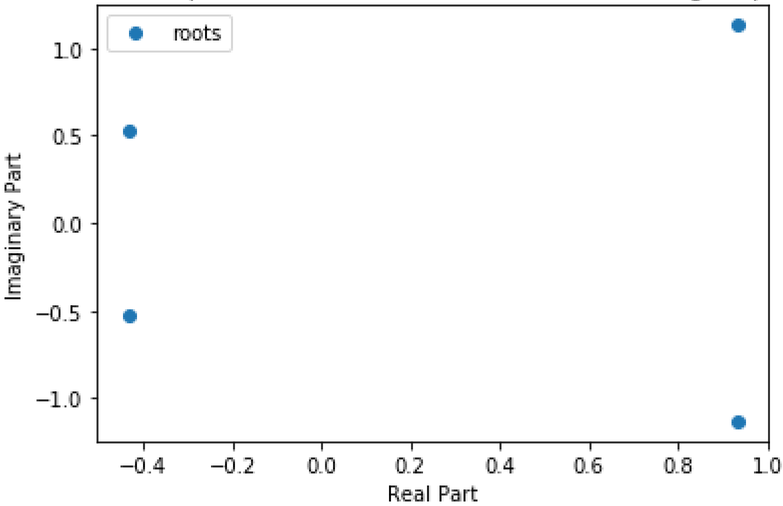
In [71]:

```
poly_root(binom, 'Binomial')
```

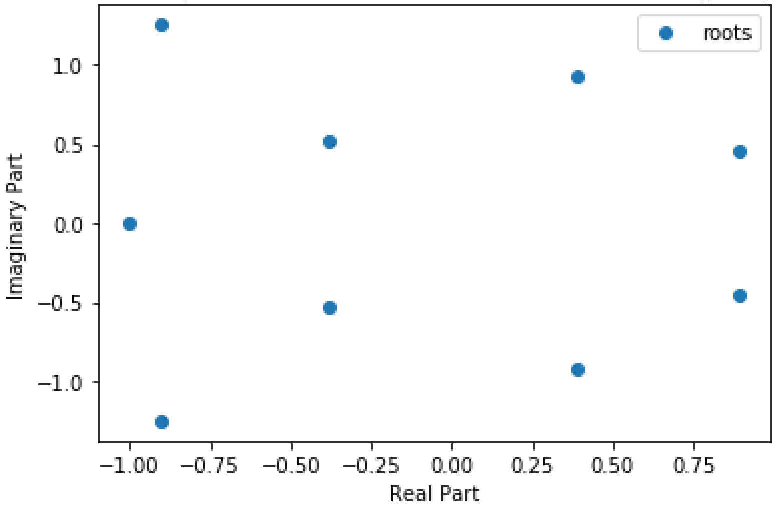
Out[71]:

```
array([1.2476, 1.8248, 2.2584, 2.7564, 3.178 ])
```

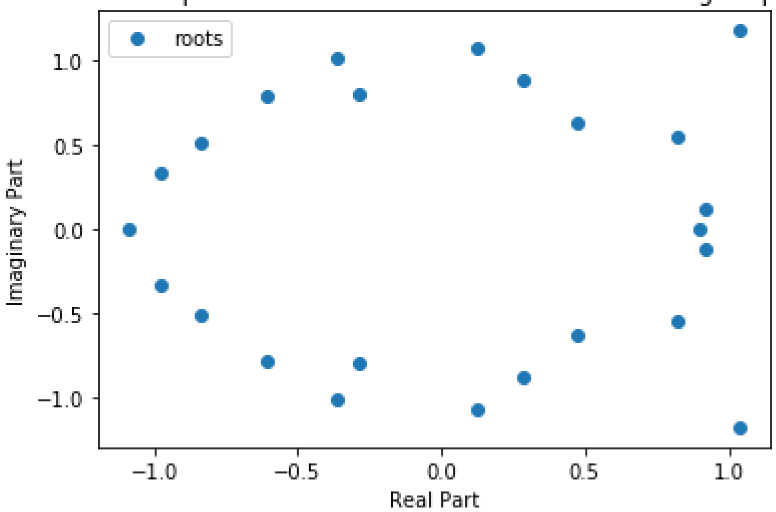
Roots in the Complex Plane for Binomial coeffecients 5 degree polynomial



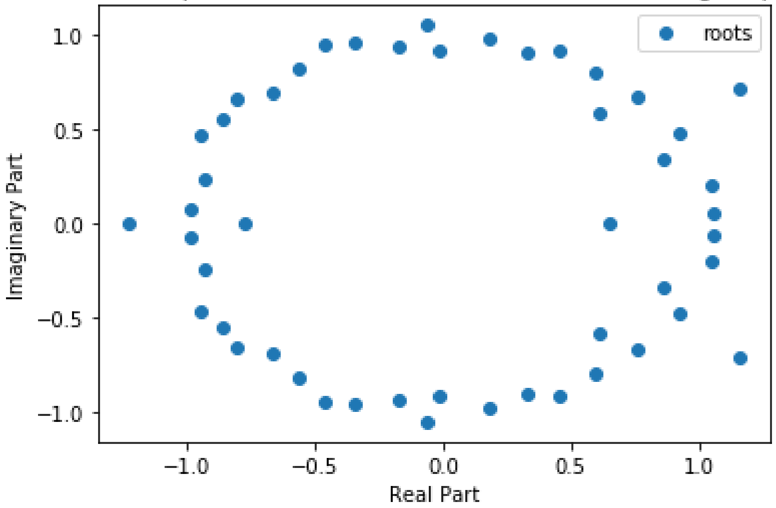
Roots in the Complex Plane for Binomial coeffecients 10 degree polynomial



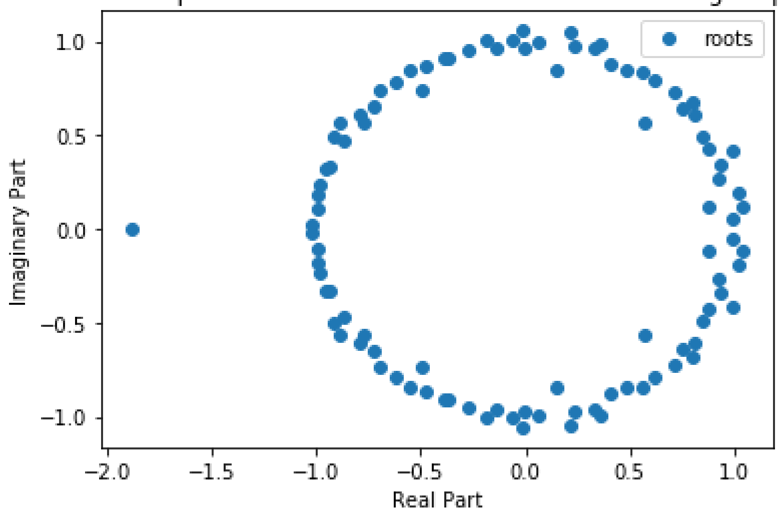
Roots in the Complex Plane for Binomial coeffecients 25 degree polynomial



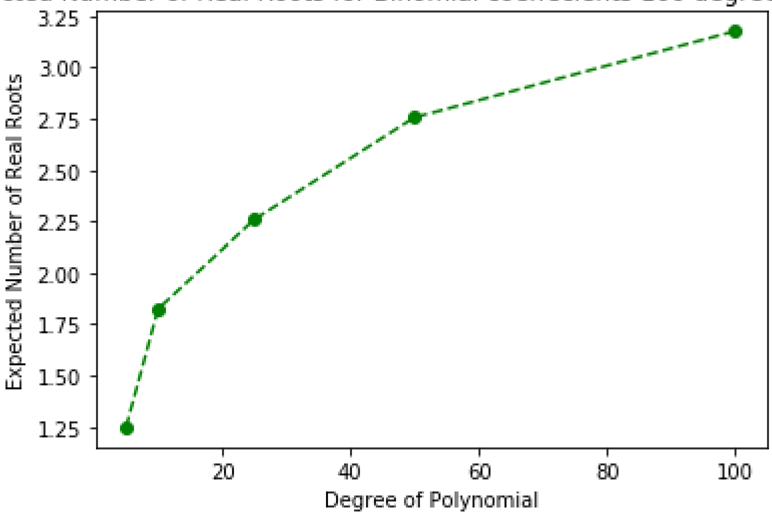
Roots in the Complex Plane for Binomial coeffecients 50 degree polynomial



Roots in the Complex Plane for Binomial coeffecients 100 degree polynomial



Expected Number of Real Roots for Binomial coeffecients 100 degree polynomial



Normal Distribution

In [20]:

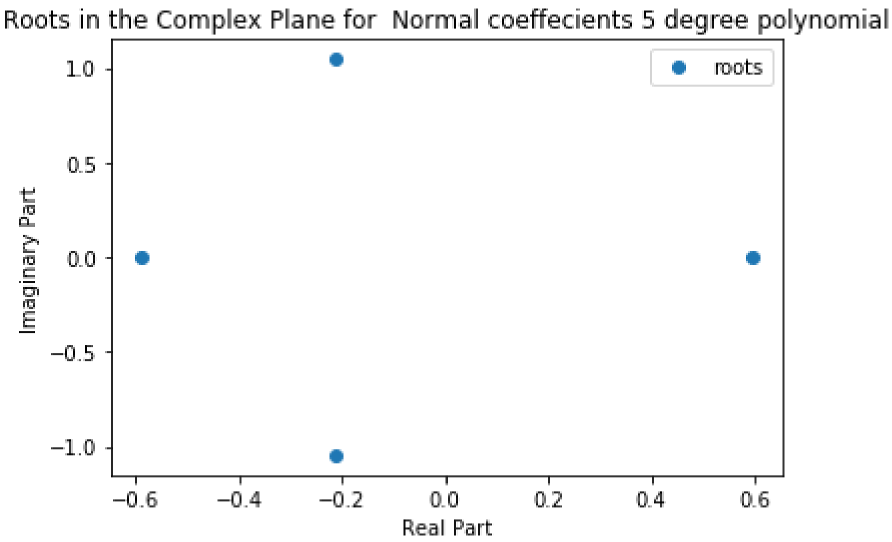
```
def normal(n):
    a = np.zeros((2, n))
    u1 = np.random.uniform(size=n)
    u2 = np.random.uniform(size=n)
    a[0] = np.sqrt(-2*np.log(u1))* np.cos(2*np.pi*u2)
    a[1] = np.sqrt(-2*np.log(u1))* np.sin(2*np.pi*u2)
    return a[0]
```

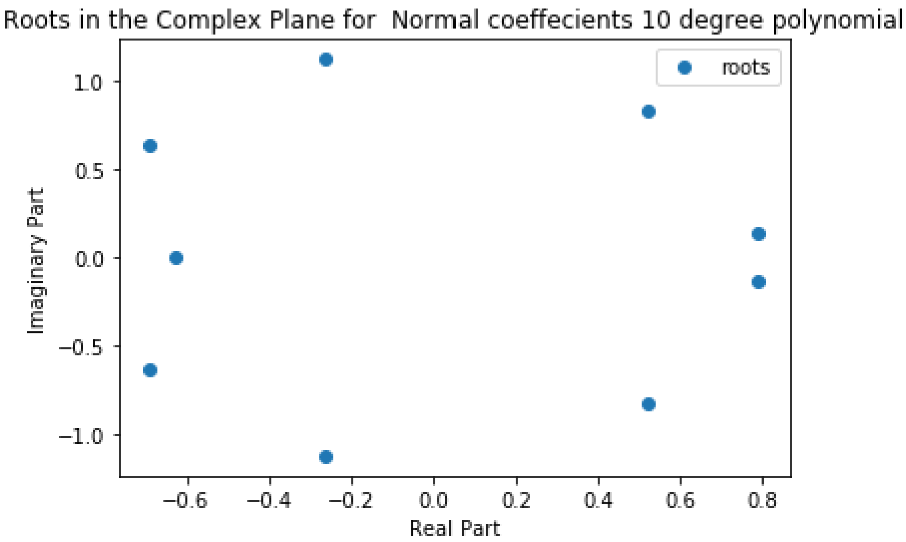
In [72]:

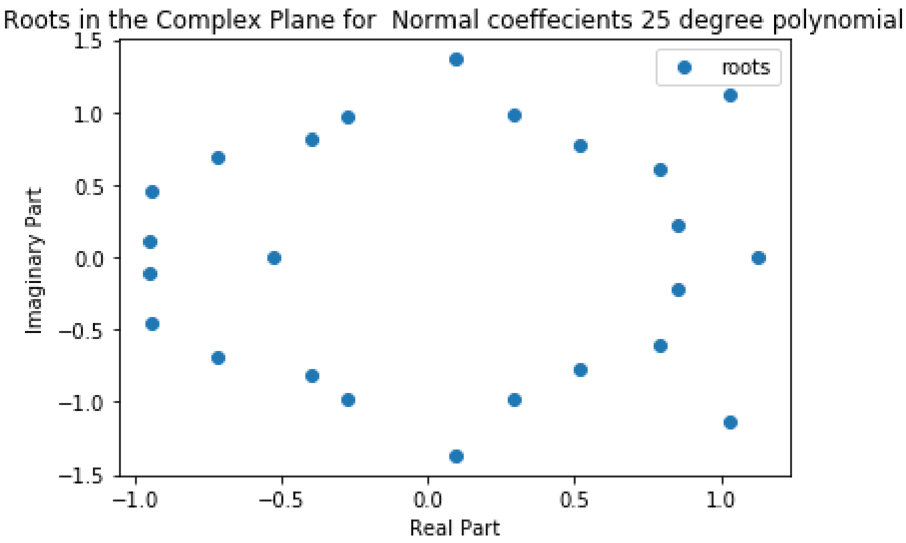
```
poly_root(normal, 'Normal')
```

Out[72]:

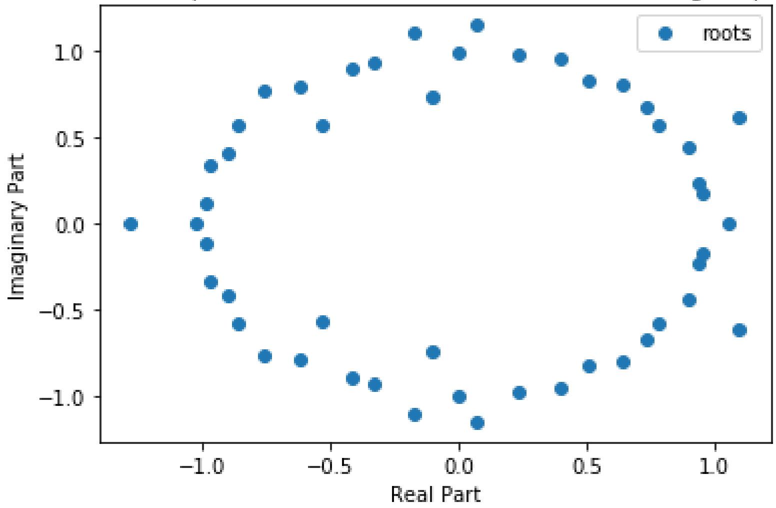
```
array([1.63  , 2.0612, 2.646 , 3.1132, 3.536 ])
```



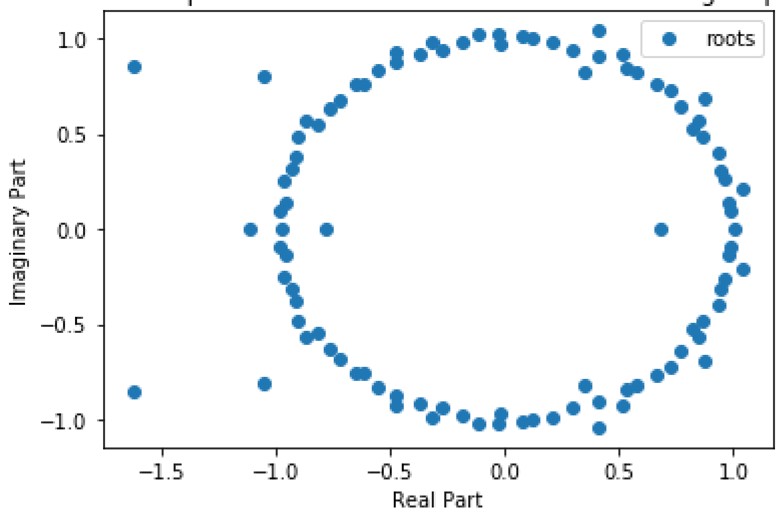




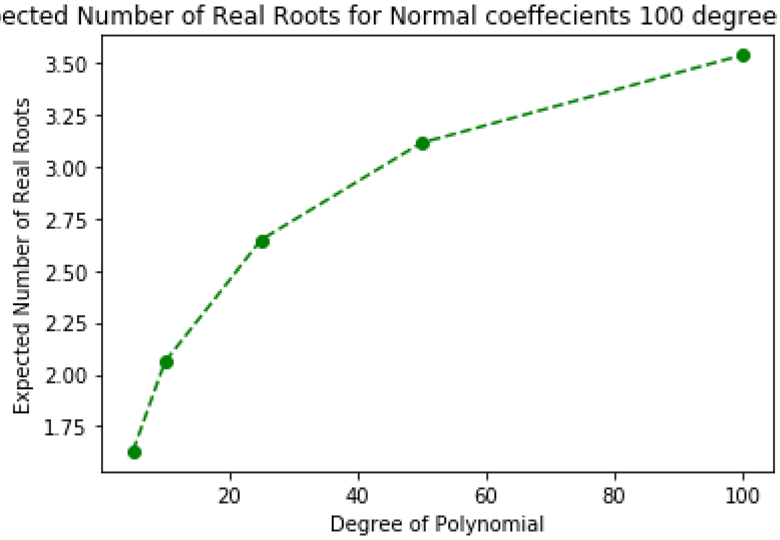
Roots in the Complex Plane for Normal coeffecients 50 degree polynomial



Roots in the Complex Plane for Normal coeffecients 100 degree polynomial



Expected Number of Real Roots for Normal coeffecients 100 degree polynomial



Cauchy Distribution

In [31]:

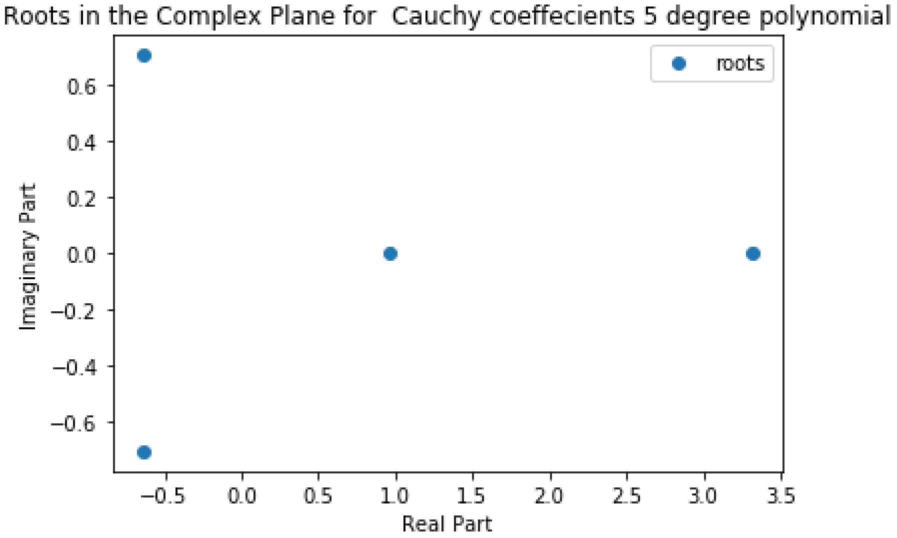
```
def cauchy(n):
    u = np.random.uniform(size=n)
    return np.tan(np.pi*(u-0.5))
```

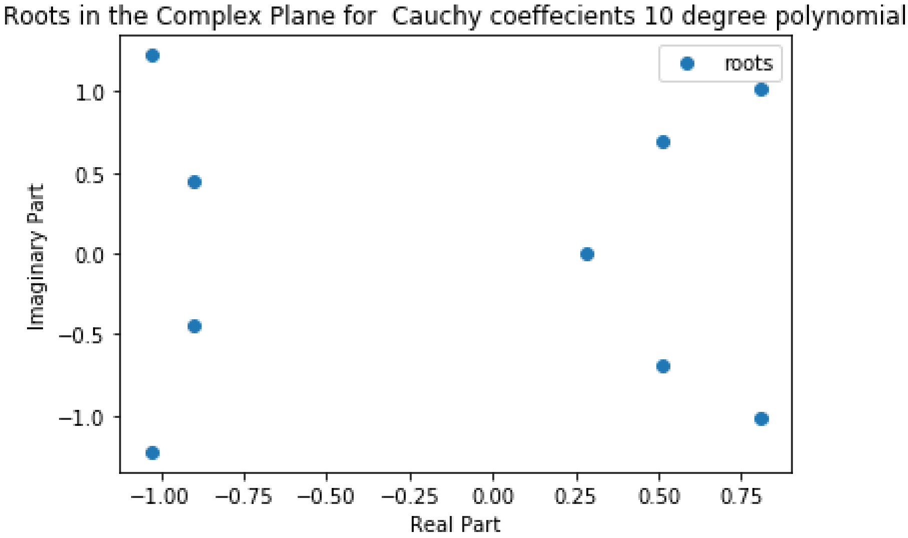
In [73]:

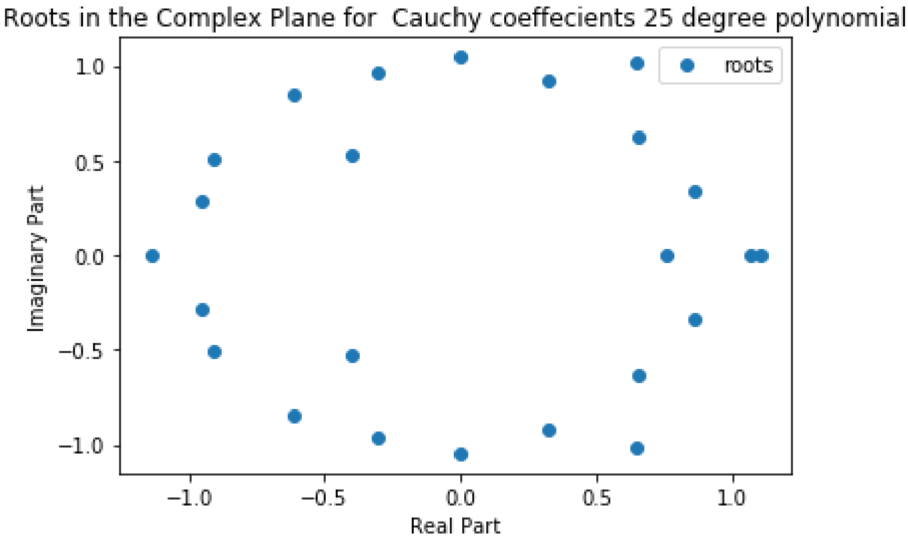
```
poly_root(normal, 'Cauchy')
```

Out[73]:

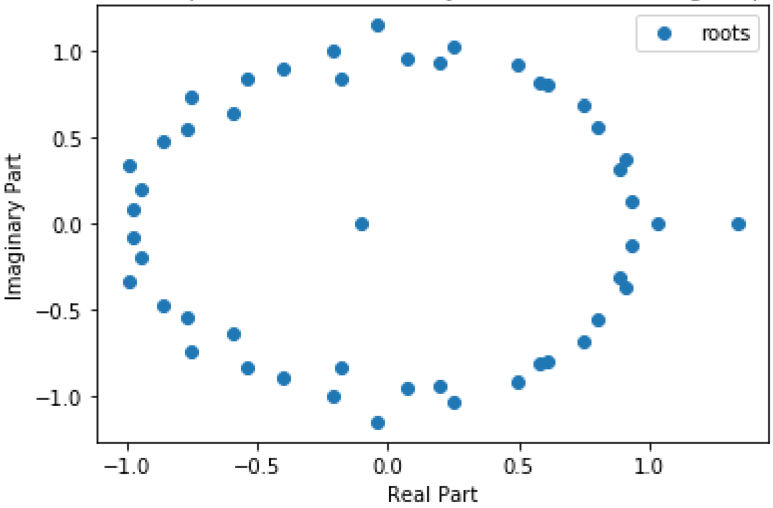
```
array([1.6584, 2.0756, 2.6676, 3.1324, 3.5644])
```



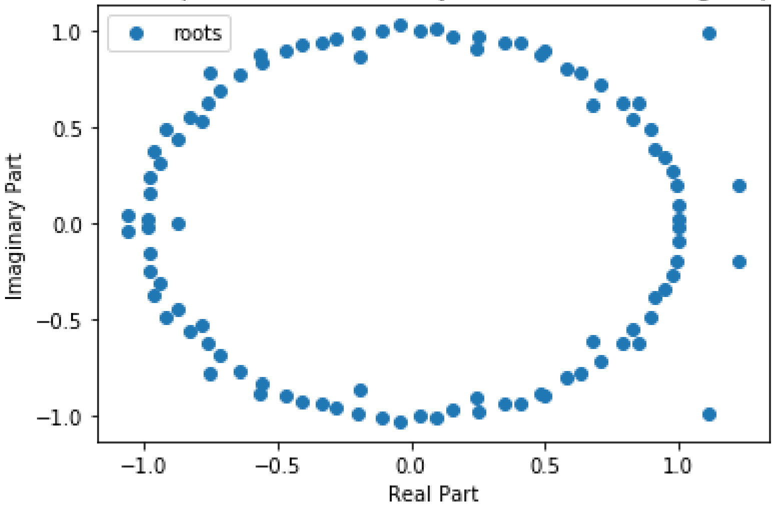




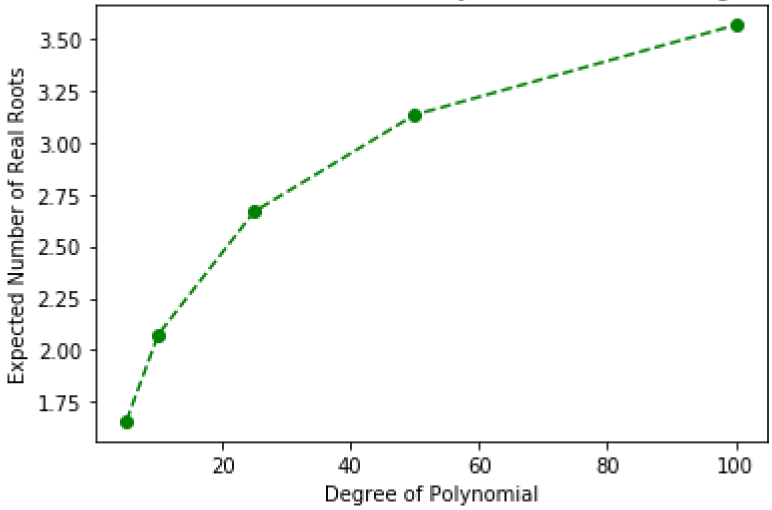
Roots in the Complex Plane for Cauchy coeffecients 50 degree polynomial



Roots in the Complex Plane for Cauchy coeffecients 100 degree polynomial



Expected Number of Real Roots for Cauchy coeffecients 100 degree polynomial



Exponential Distribution

In [39]:

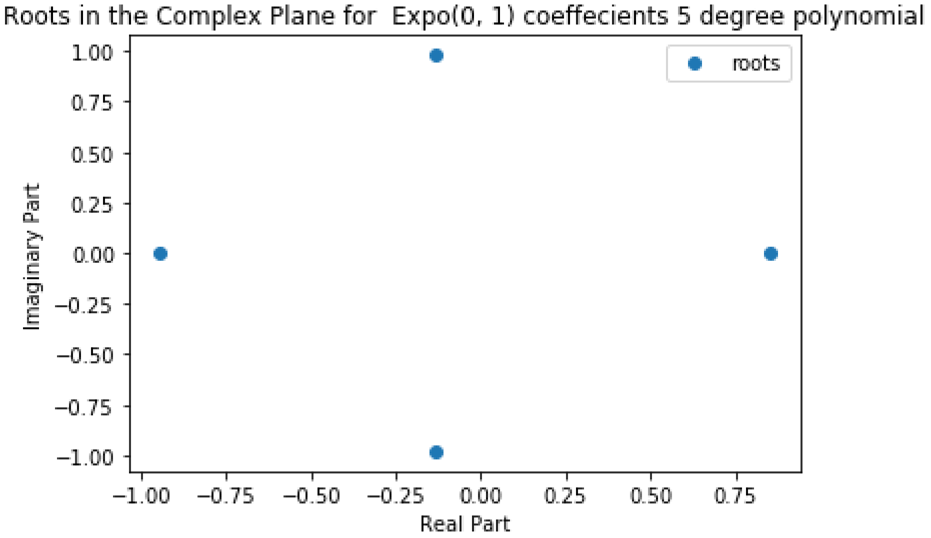
```
def expon(n):
    u = np.random.uniform(size=n)
    return -np.log(u)
```

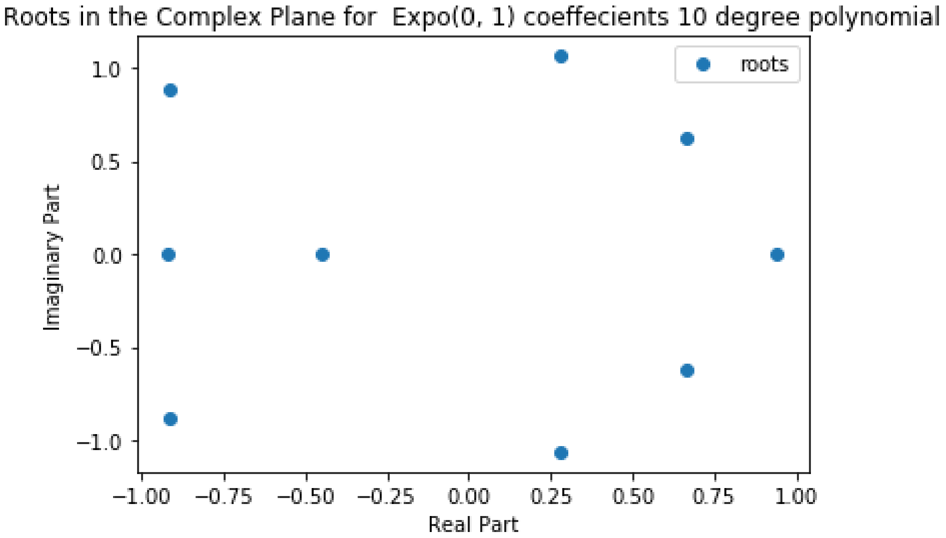
In [74]:

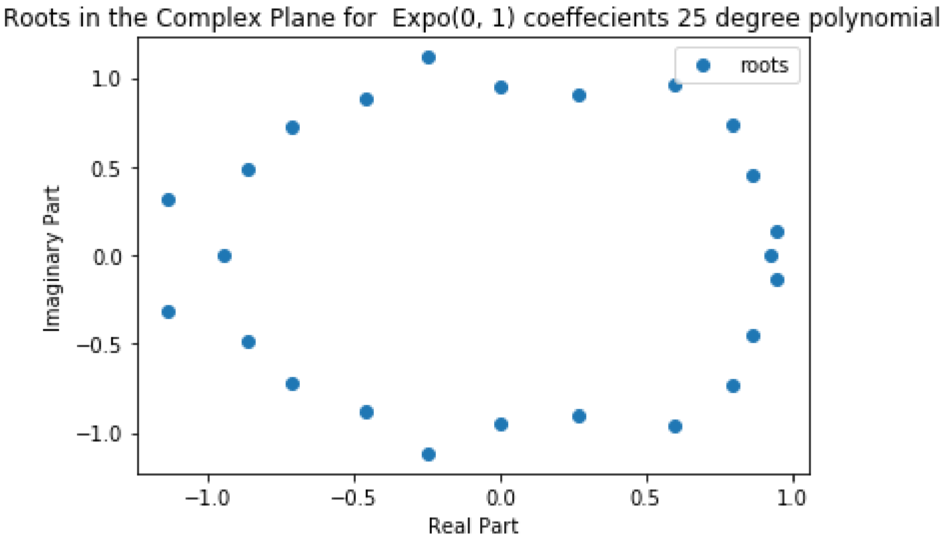
```
poly_root(normal, 'Expo(0, 1)')
```

Out[74]:

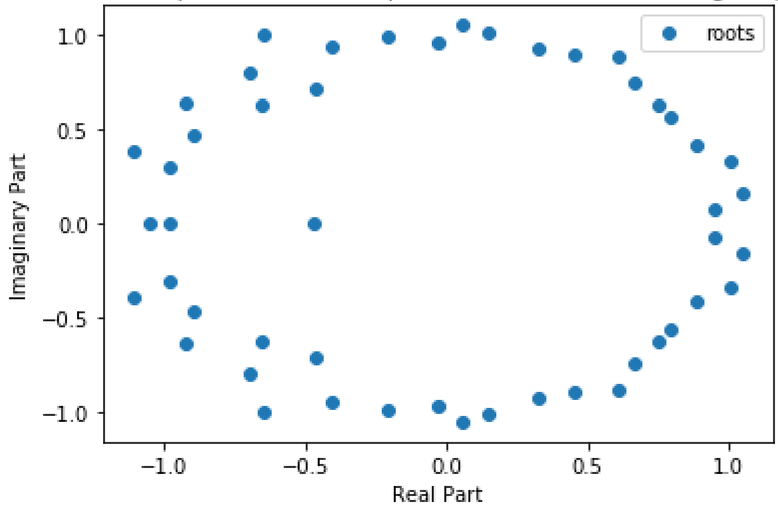
```
array([1.6576, 2.0848, 2.66  , 3.1416, 3.5444])
```



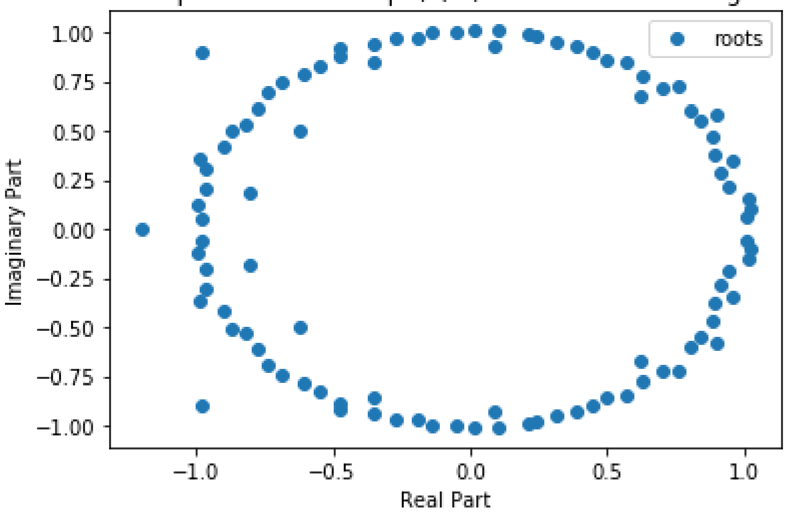




Roots in the Complex Plane for Expo(0, 1) coeffecients 50 degree polynomial



Roots in the Complex Plane for Expo(0, 1) coeffecients 100 degree polynomial



Expected Number of Real Roots for Expo(0, 1) coeffecients 100 degree polynomial