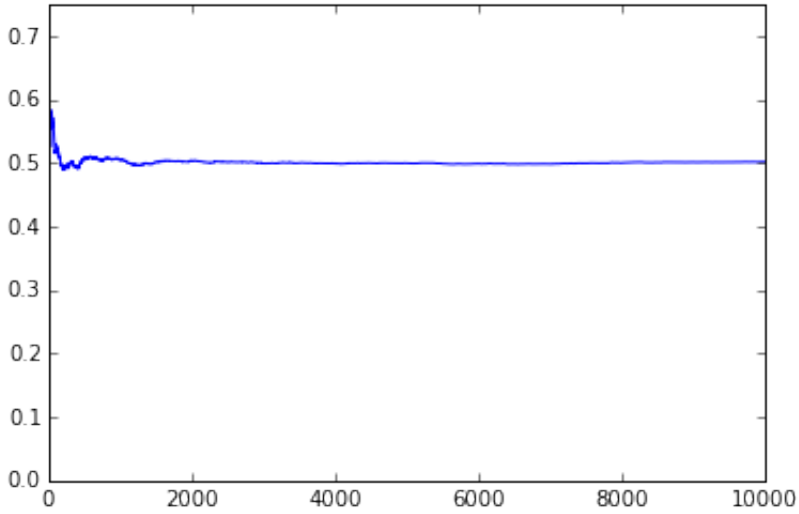
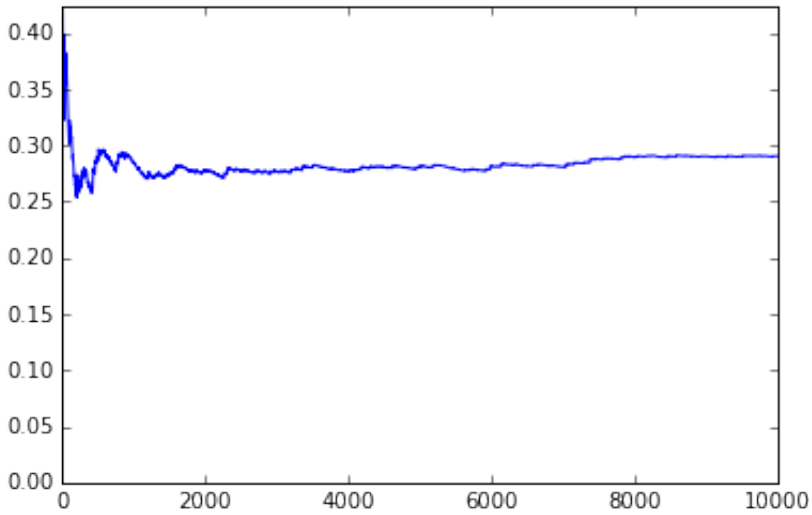
In [10]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
import math
%matplotlib inline
```

In [49]:
```
N = 10000
numbers = np.arange(1, N)

def eval(n, k, X):
    moment = (X**k).sum()/(n*1.0)
    return ((math.factorial(k)*1.0)/moment)**(1.0/k)

errors = []
distr = sts.expon(1.0)
bigSample = distr.rvs(N)

def go(k, scale):
    y = np.array([])
    for n in numbers:
        sample = bigSample[:n]
        y = np.append(y, abs(eval(n, k, sample) - 1.0))
    print '\n\n'
    print 'n=10^4, error for k = ', k, ' : ', y[-1]
    errors.append(y[-1])
    plt.plot(numbers, y, label='k = '+str(k))
    plt.ylim(0, y.mean()*1.5)
    plt.show()

for k in np.arange(1, 20, 2):
    go(k, 1.0/(10.0*k))

print '\n\n\n\n'
print 'best k = ',errors.index(min(errors))+1,' error is ',min(erro
rs)
```

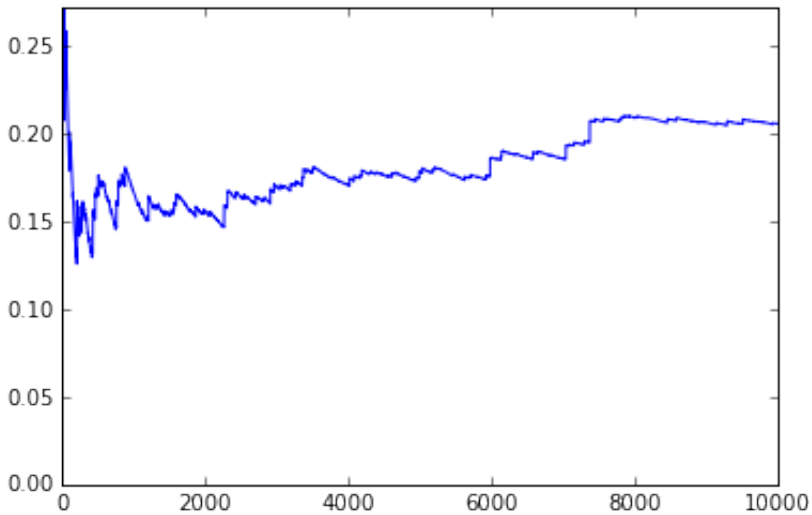
```
n=10^4, error for k =  1  :  0.501972706496
```



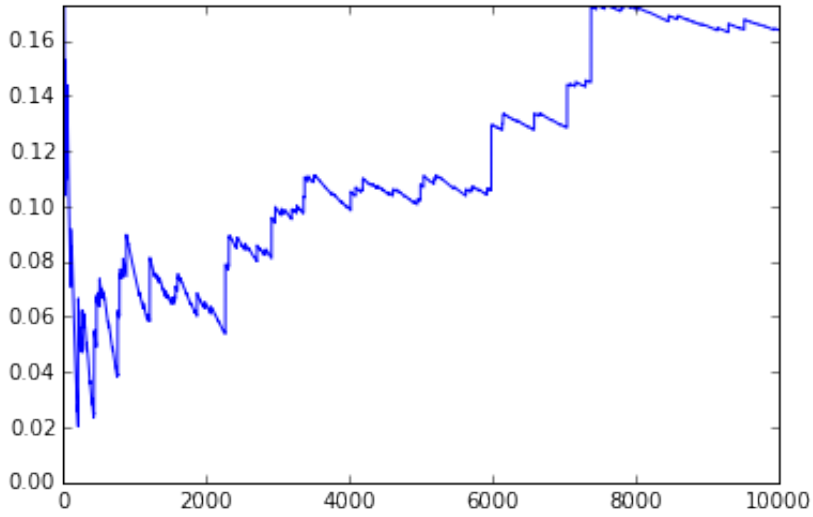

```
n=10^4, error for k =  3  :  0.29042994528
```



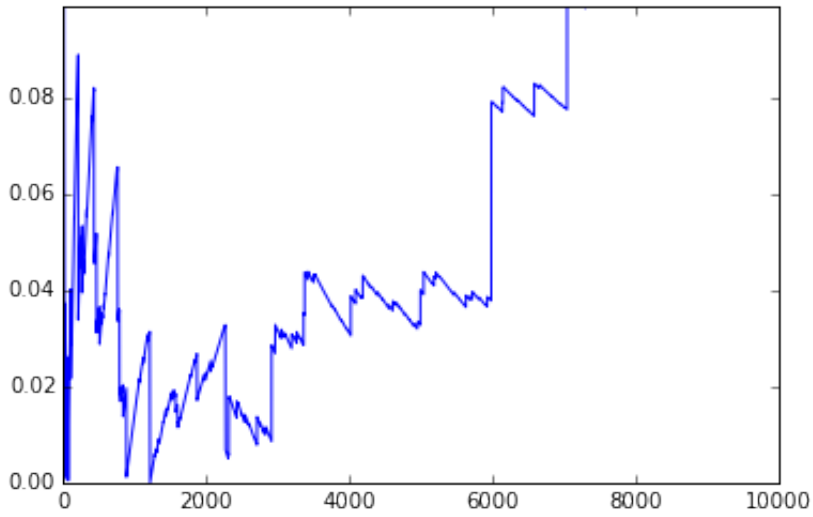

```
n=10^4, error for k =  5  :  0.205916018258
```



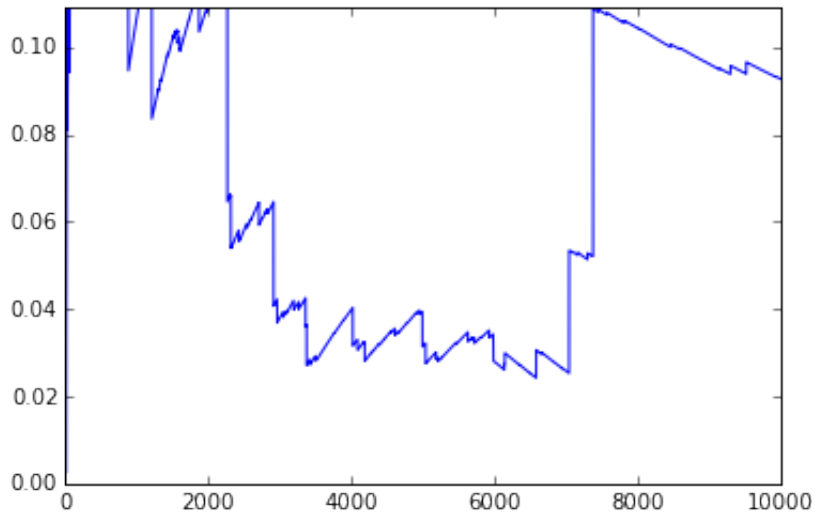
```
n=10^4, error for k =  7  :  0.163821336931
```



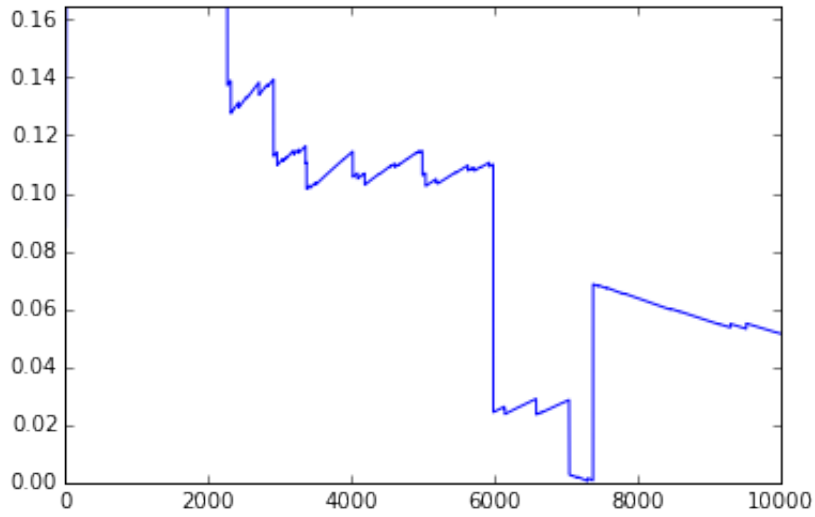

```
n=10^4, error for k =  9  :  0.129391487906
```



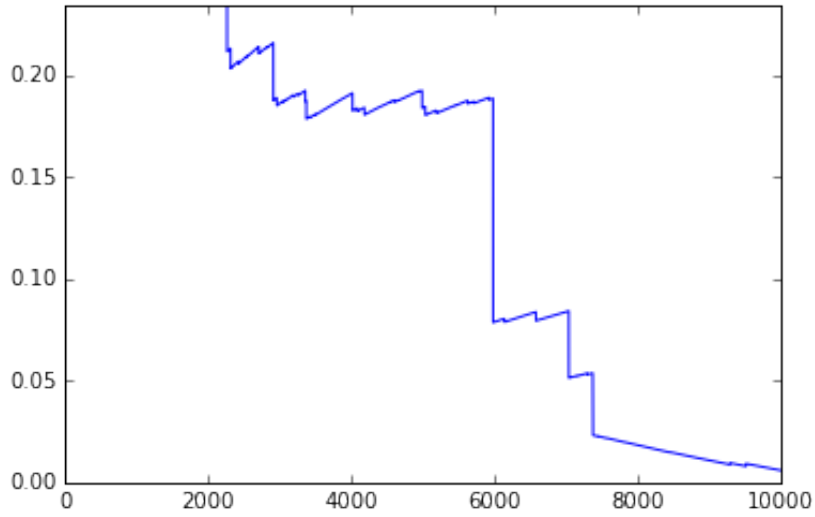

```
n=10^4, error for k =  11  :  0.0926297055921
```



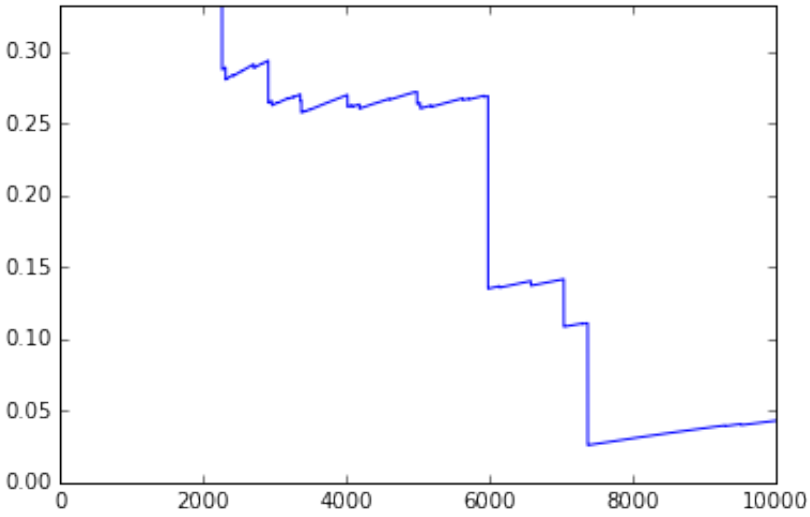

```
n=10^4, error for k =  13  :  0.0514380976545
```



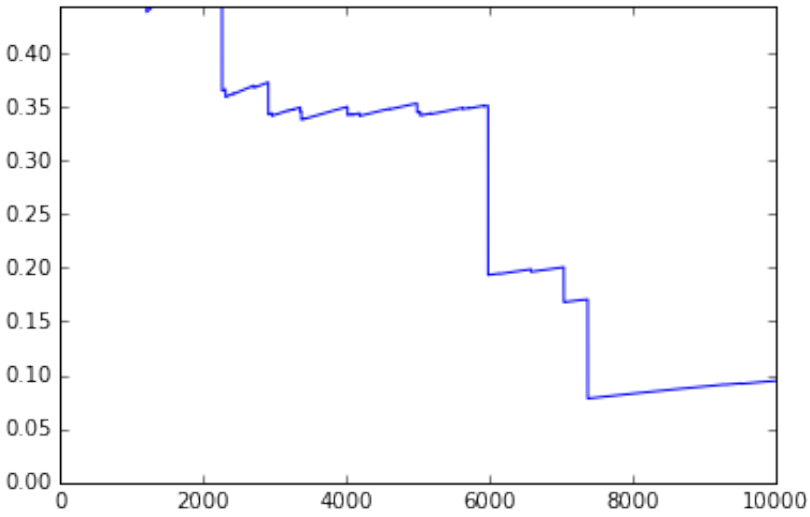

```
n=10^4, error for k =  15  :  0.00597397527767
```



```
n=10^4, error for k =  17  :  0.0430396568605
```




```
n=10^4, error for k =  19  :  0.0948440376919
```




```
best k =  8  error is  0.00597397527767
```

In [ ]: