

```
In [142]: import scipy.stats as sts
import scipy.misc as sm
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Сгенерируйте выборку X_1, \dots, X_N из экспоненциального распределения с параметром $\theta = 1$ для $N = 10^4$.

```
In [143]: theta = 1.
samples = sts.expon.rvs(size=10000, loc=0, scale=theta)
N = np.size(samples)
```

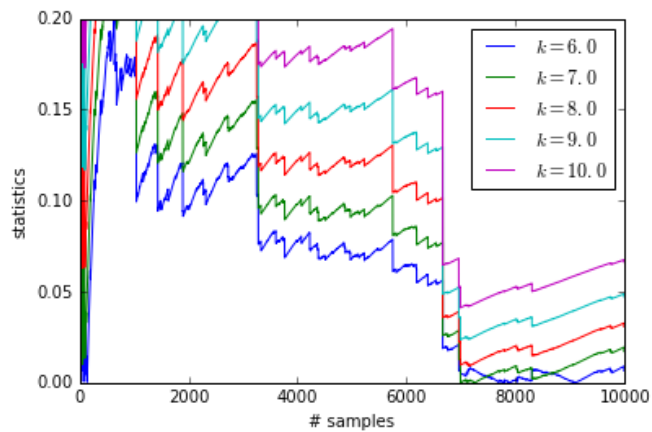
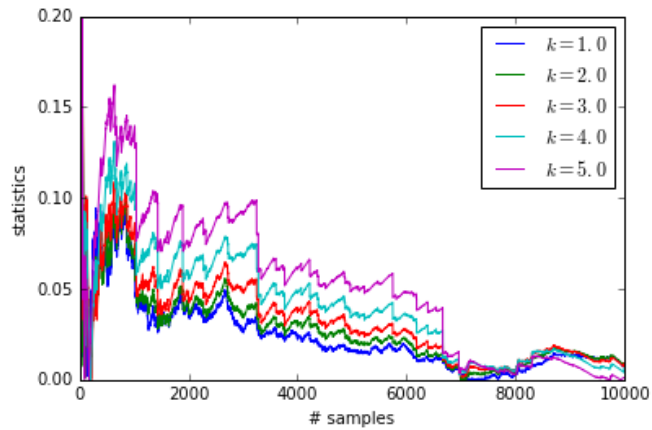
Для всех $n \leq N$ посчитайте оценку $(k!/X^n k)^{1/k}$ параметра $\theta = 1$.

```
In [144]: k = np.arange(10) + 1.
res = [[] for x in range(10)]

for i in range(10):
    samples2 = samples ** k[i]
    for n in range(1, N + 1) :
        ans = (sm.factorial(k[i]) / np.mean(samples2[:n])) ** (1. / k[i])
        res[i].append(np.fabs(ans - theta))
```

Проведите исследование, аналогичное предыдущей задаче, и выясните, при каком k оценка ведет себя лучше (рассмотрите не менее 10 различных значений k).

```
In [145]: colors = ['red', 'blue', 'green', 'black', 'purple']
x = np.arange(N) + 1.
for j in range(2):
    plt.xlabel("# samples")
    plt.ylabel("statistics")
    plt.ylim([0., 0.2])
    plt.xlim([1., N])
    for i in range(5):
        plt.plot(x, res[i + j * 5], label="$k = " + str(k[i + j * 5]) + "$")
    plt.legend()
    plt.show()
```



При всех k при увеличении выборки оценка становится ближе к тета. Но все-таки самая лучшая оценка в большинстве случаев получается при $k = 1$.

In []: