

1.2

March 11, 2016

1 Задача 2

```
In [14]: %matplotlib inline
import numpy as np
import math as mt
import matplotlib
import matplotlib.pyplot as plt
from pylab import *
from scipy.stats import *

In [15]: import random

# Рандомный цвет
def RandCol():
    r = lambda: random.randint(0,255)
    return '%02X%02X%02X' % (r(),r(),r())

In [16]: # Построение графиков для всех натуральных k, принадлежащих
# отрезку [fr, to], h - ограничение графика по высоте
def Exponent(fr, to, h):
    N = 10000

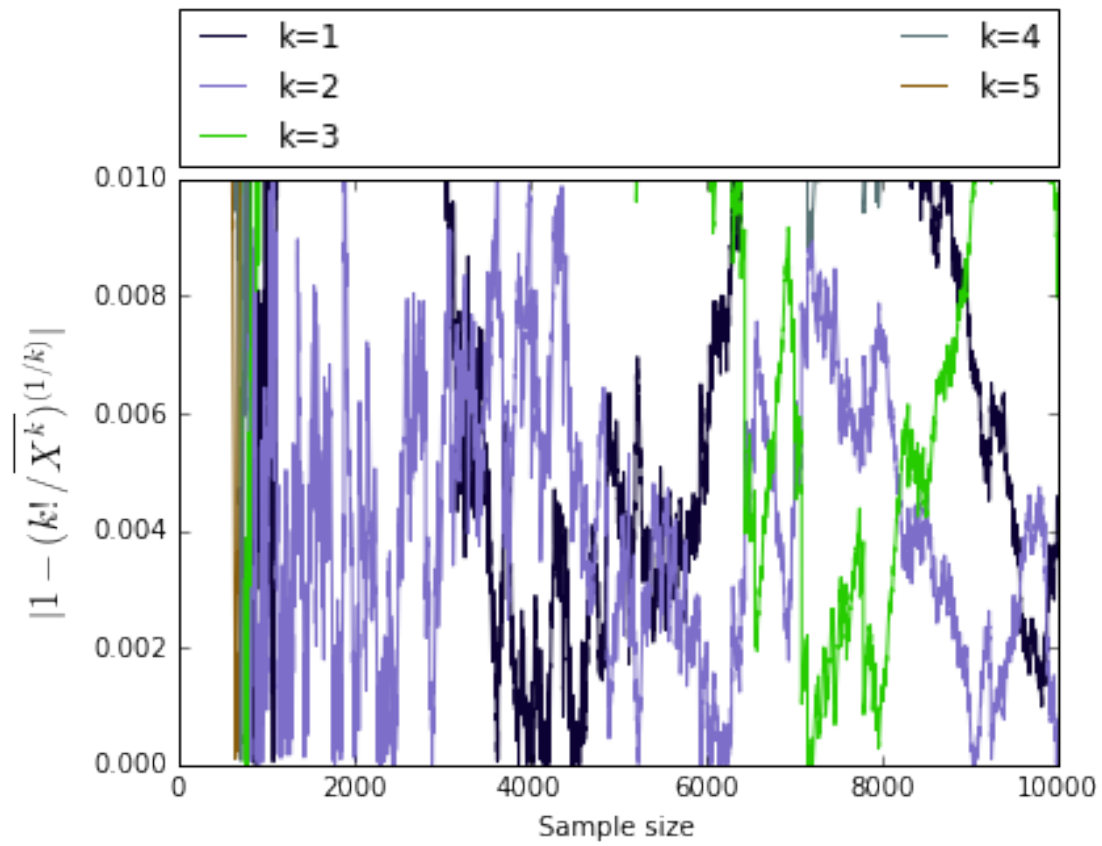
    # Выборка из экспоненциального распределения размера N
    s_exp = expon.rvs(size=N)

    est = np.zeros((to-fr+1, N-1))
    for k in range(fr, to+1):
        # Заполнение массива est оценками параметра theta
        est[k-fr] = np.array([(mt.factorial(k)/(sum(s_exp[:n]**k)/n))**(1/float(k))\
                               for n in range(1,N)])

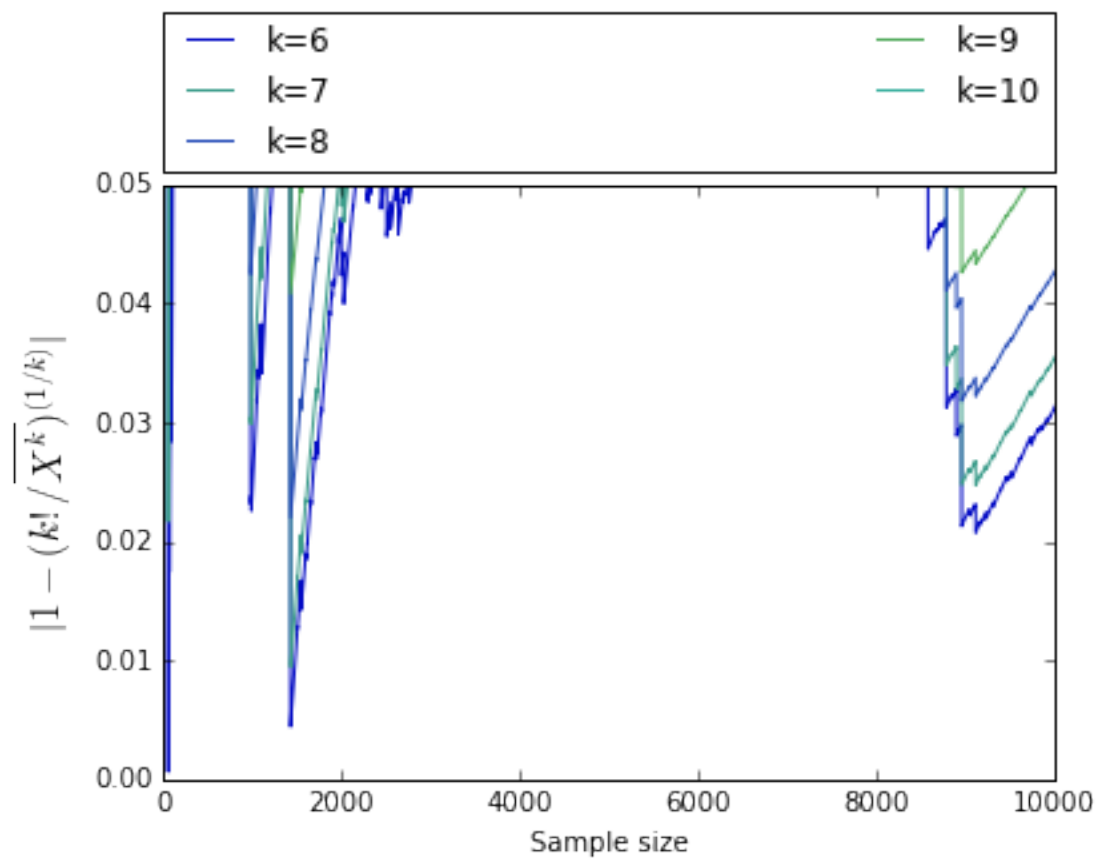
    # Построение графиков
    figure()
    ylim(0, h)
    xlabel('Sample size')
    ylabel('$|1-(k!/\overline{X^k})^{(1/k)}|$', fontsize=15)

    for k in range(fr, to+1):
        plot(abs(est[k-fr]-1), 'r', label='k={}'.format(k), color=RandCol())
    legend(bbox_to_anchor=(0., 1.02, 1., .102), loc=3, ncol=2, \
           mode="expand", borderaxespad=0.)
    show()
```

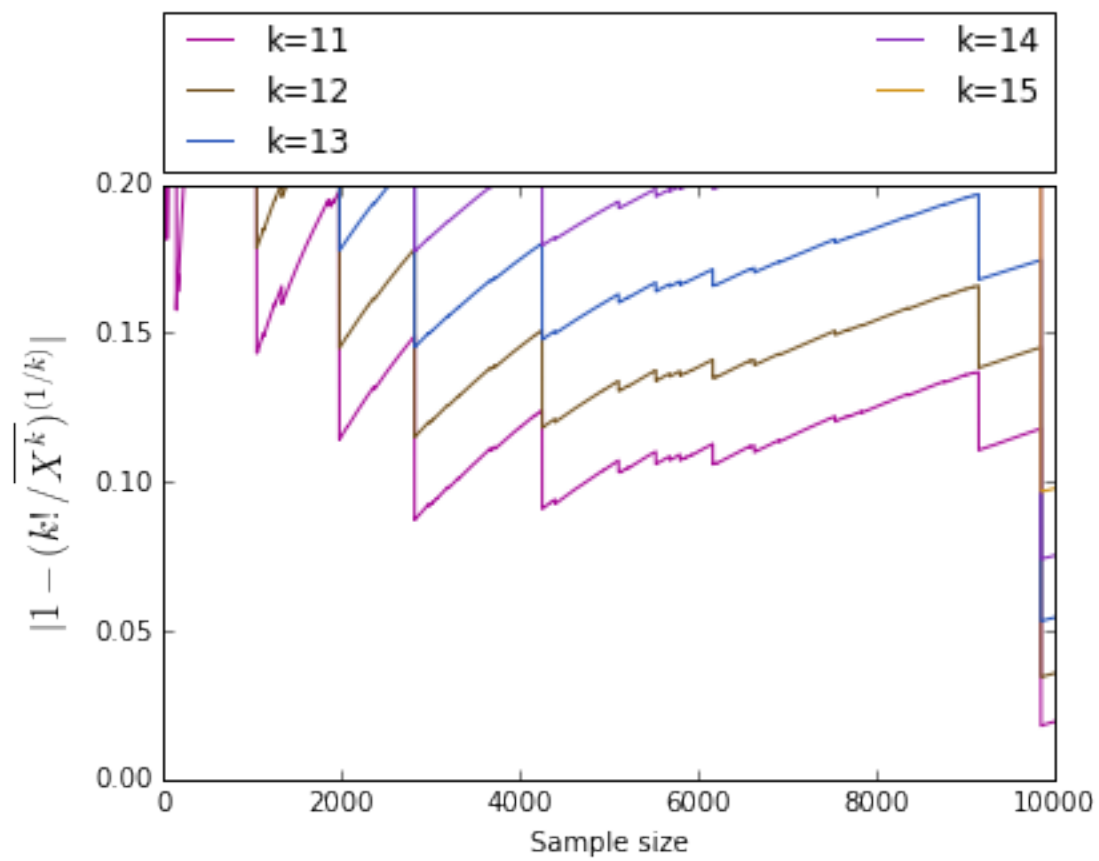
In [17]: `Exponent(1, 5, 0.01)`



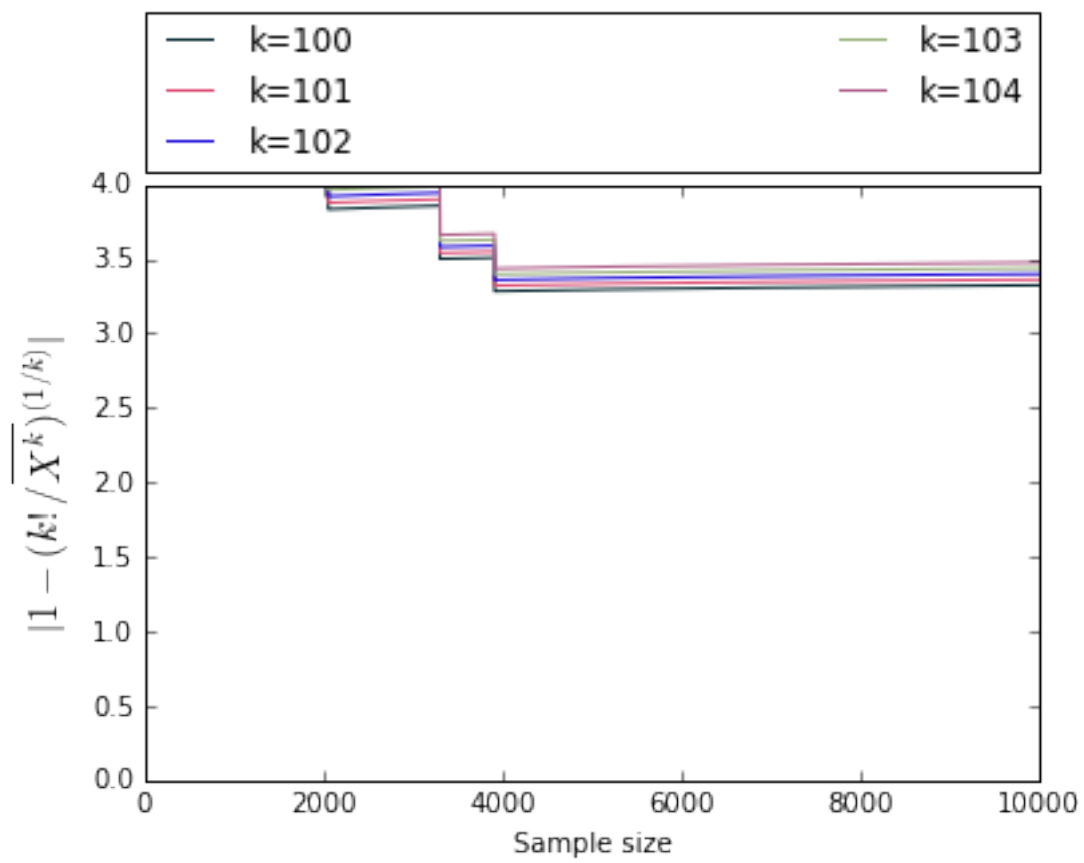
In [12]: `Exponent(6, 10, 0.05)`



In [18]: `Exponent(11,15,0.2)`



In [20]: `Exponent(100,104,4)`



Опыт показал, что при $k \in \{1, 2\}$ оценка ведет себя лучше всего.