Задача 2.2

In [11]:

```
%matplotlib inline
import scipy.stats as st
import numpy as np
import matplotlib.pyplot as plt
import math
```

Генерируем выборку из экспоненциального распределения с параметром $\theta = 1$ размера $N = 10000$

In [12]:

```
teta = 1.0 #параметр распределения
expon = st.expon(loc=0, scale=1.0/teta)
a = expon.rvs(size=10000)
```

```
Для каждого $n < N$ считаем оценку $\left(\frac{k!}{\overline{X^{k}}}\right)^ {(1/k)}$
параметра $\theta$ = 1.
(делаю не в цикле потому, что очень долго считает)
```

In [26]:

```
k=1

fact = math.factorial(k)
stat2 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat2[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat2[10000]))
```

```
Значение оценки на всей выбоке: 1.0130582387777003
```

In [13]:

```
k=2

fact = math.factorial(k)
stat0 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat0[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat0[10000]))
```

```
Значение оценки на всей выбоке:1.0185616405960236
```

In [14]:

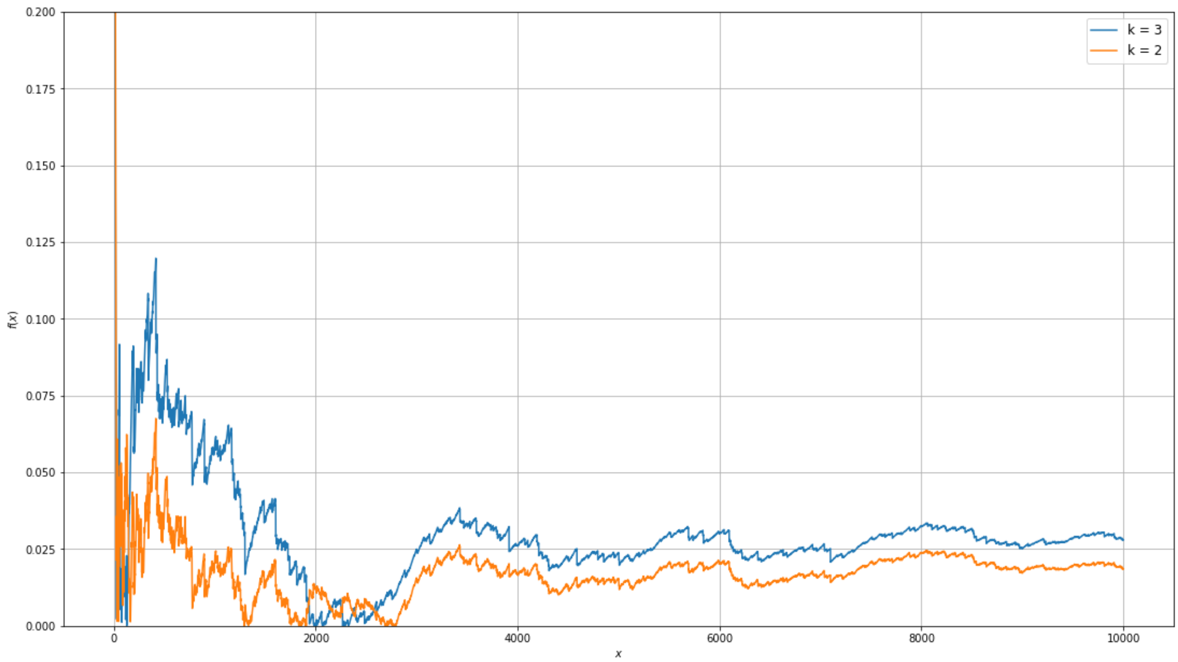
```
k=3

fact = math.factorial(k)
stat1 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat1[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выборке: " + str(stat1[10000]))
```

Значение оценки на всей выборке: 1.0280123772382364

Посмотрим на график при $k = 2, 3$. Чтобы убедиться, что все делаем правильно.

In [15]:

```
plt.plot(np.arange(1,10001), abs(stat1 - teta)[1:], label =r'k = 3')
plt.plot(np.arange(1,10001), abs(stat0 - teta)[1:], label =r'k = 2')
plt.xlabel(r'$x$') #Метка по оси x в формате TeX
plt.ylabel(r'$f(x)$') #Метка по оси y в формате TeX
plt.legend(fontsize=12, loc=1)
plt.grid(True) #Сетка
fig = plt.gcf()
plt.ylim (0, 0.2)
fig.set_size_inches(18.5, 10.5)
plt.show()
```



Замечаем, что при $k = 2$, оценка себя ведет лучше(отклоняется от нуля меньше), чем при $k = 3$.

In [18]:

```
k=4

fact = math.factorial(k)
stat3 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat3[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat3[10000]))
```

Значение оценки на всей выбоке: 1.04123392685858

In [19]:

```
k=5

fact = math.factorial(k)
stat4 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat4[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat4[10000]))
```

Значение оценки на всей выбоке: 1.058610121355865

In [20]:

```
k=6

fact = math.factorial(k)
stat5 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat5[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat5[10000]))
```

Значение оценки на всей выбоке: 1.0801587955732488

In [21]:

```
k=7

fact = math.factorial(k)
stat6 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat6[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat6[10000]))
```

Значение оценки на всей выбоке: 1.105426592892592

In [22]:

```
k=8

fact = math.factorial(k)
stat7 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat7[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat7[10000]))
```

Значение оценки на всей выбоке: 1.1337323925264344

In [23]:

```
k=9

fact = math.factorial(k)
stat8 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat8[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat8[10000]))
```

Значение оценки на всей выбоке: 1.1643957008523604

In [24]:

```
k=10

fact = math.factorial(k)
stat9 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat9[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat9[10000]))
```

Значение оценки на всей выбоке: 1.1968529448793572

In [32]:

```
k=30

fact = math.factorial(k)
stat30 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat30[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выбоке: " + str(stat30[10000]))
```

Значение оценки на всей выбоке: 1.9909149372470463

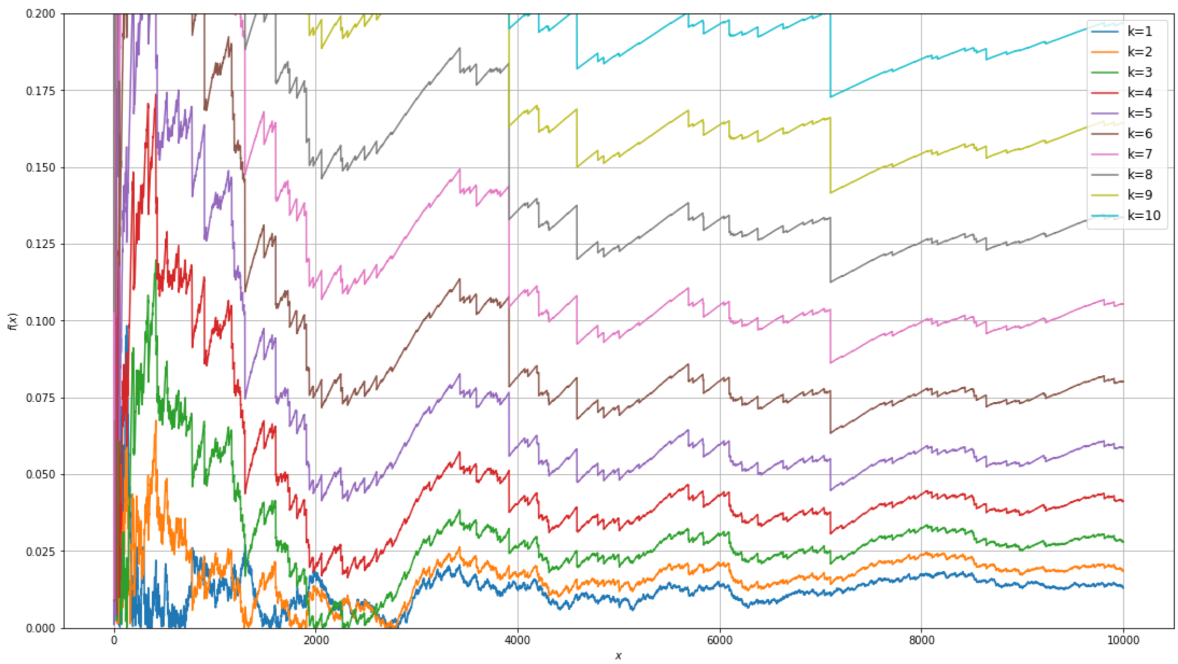
In [35]:

```
k=100

fact = math.factorial(k)
stat100 = np.zeros(10001)
for i in range(1,10001):
    m_k = [(z)**(k) for z in a[:i]]
    mom_k = np.sum(m_k)/i
    stat100[i] = math.pow((fact/mom_k),(1.0/k))
print ("Значение оценки на всей выборке: " + str(stat100[10000]))
```

Значение оценки на всей выборке: 5.156662765502462

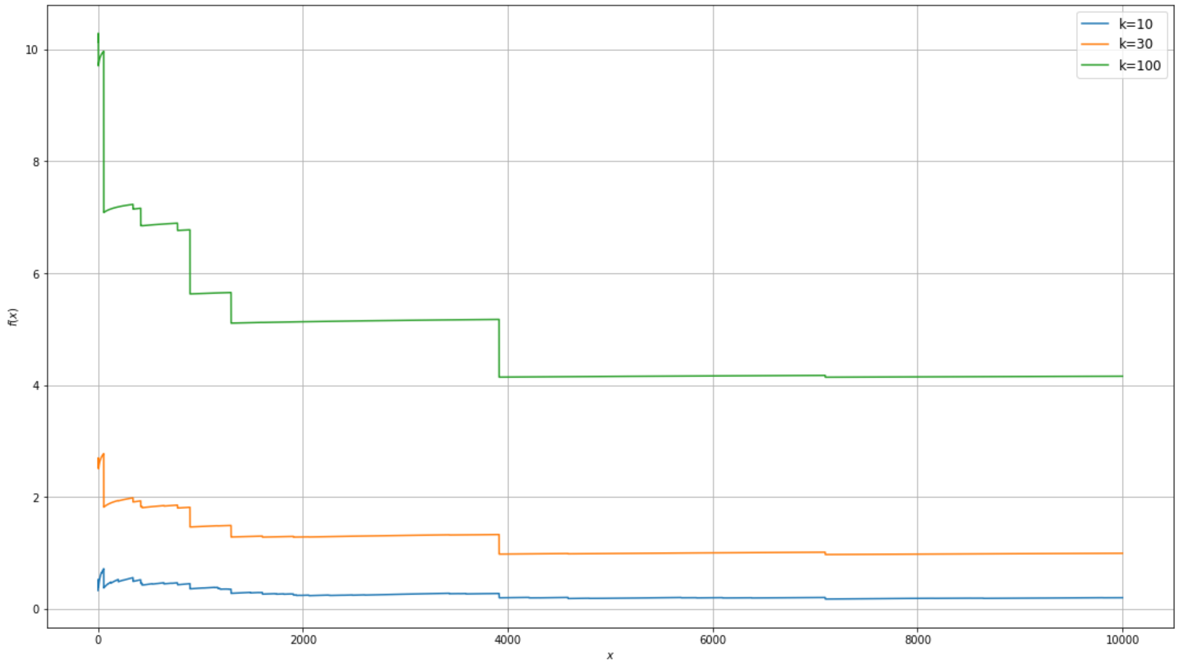
In [31]:

```
plt.plot(np.arange(1,10001), abs(stat2 - teta)[1:], label  = r'k=1')
plt.plot(np.arange(1,10001), abs(stat0 - teta)[1:], label  = r'k=2')
plt.plot(np.arange(1,10001), abs(stat1 - teta)[1:], label  = r'k=3')
plt.plot(np.arange(1,10001), abs(stat3 - teta)[1:], label  = r'k=4')
plt.plot(np.arange(1,10001), abs(stat4 - teta)[1:], label  = r'k=5')
plt.plot(np.arange(1,10001), abs(stat5 - teta)[1:], label  = r'k=6')
plt.plot(np.arange(1,10001), abs(stat6 - teta)[1:], label  = r'k=7')
plt.plot(np.arange(1,10001), abs(stat7 - teta)[1:], label  = r'k=8')
plt.plot(np.arange(1,10001), abs(stat8 - teta)[1:], label  = r'k=9')
plt.plot(np.arange(1,10001), abs(stat9 - teta)[1:], label  = r'k=10')
plt.legend(fontsize=12, loc=1)
plt.xlabel(r'$x$') #Метка по оси x в формате TeX
plt.ylabel(r'$f(x)$') #Метка по оси y в формате TeX
#plt.title(r'$y=x^2$') #Заголовок в формате TeX
plt.grid(True) #Сетка
fig = plt.gcf()
plt.ylim (0, 0.2)
fig.set_size_inches(18.5, 10.5)
plt.show()
```



In [36]:

```
plt.plot(np.arange(1,10001), abs(stat9 - teta)[1:], label  = r'k=10')
plt.plot(np.arange(1,10001), abs(stat30 - teta)[1:], label  = r'k=30')
plt.plot(np.arange(1,10001), abs(stat100 - teta)[1:], label  = r'k=100')
plt.legend(fontsize=12, loc=1)
plt.xlabel(r'$x$') #Метка по оси x в формате TeX
plt.ylabel(r'$f(x)$') #Метка по оси y в формате TeX
#plt.title(r'$y=x^2$') #Заголовок в формате TeX
plt.grid(True) #Сетка
fig = plt.gcf()

fig.set_size_inches(18.5, 10.5)
plt.show()
```



На графике видно, что с ростом k, ухудшается оценка

Вывод

При всех k при увеличении выборки оценка становится ближе к θ. Что может объясняться тем, что оценка $\left(\frac{k!}{X^k}\right)^{(1/k)}$ является асимтотически нормальной оценкой θ для любого k. (Доказывали на семинаре). Заметим, что с ростом k от 1 до 10 оценка ведет себя хуже и хуже, и достигает наименьшего отклонения от теоретического значения θ при $\theta = 1$.

In [ ]: