

## Задача 2.2

In [1]:

```
import numpy as np
import math as mth
import scipy.stats as sps
import matplotlib.pyplot as plt
```

```
N=10000
%pylab inline
```

Populating the interactive namespace from numpy and matplotlib

### Вычисление оценки

In [2]:

```
def my_evaluate(sample, k):
    k_deg = sample**k;
    return (mth.factorial(k)/mean(k_deg))**(1/k)
```

### Построение графиков для разных k

Для наглядности построю 2 графика разного масштаба, чтобы было хорошо видно все линии.

In [3]:

```
sample = sps.expon.rvs(size=N)

plt.figure(figsize=(15, 25))
grid = np.arange(1, N)
colors = ['blue', 'green', 'cyan', 'black', 'pink', 'brown', 'red', 'orange',
'purple', 'yellow']
values = [1, 2, 5, 6, 10, 15, 25, 37, 50, 70]
i = 0;
ans = list(grid)
min = 1000
mink = 0;

for k in values:
    for n in grid:
        ans[n-1] = abs(my_evalue(sample[:n], k)-1)
    plt.subplot(211)
    plt.plot(grid, ans, color=colors[i], linewidth=1, label='k = '+ str(values
[i]))
    plt.subplot(212)
    plt.plot(grid, ans, color=colors[i], linewidth=1, label='k = '+ str(values
[i]))
    if (ans[-1] < min):
        min = ans[-1]
        mink = k
    i+=1

plt.legend()
plt.xlim((1, N))

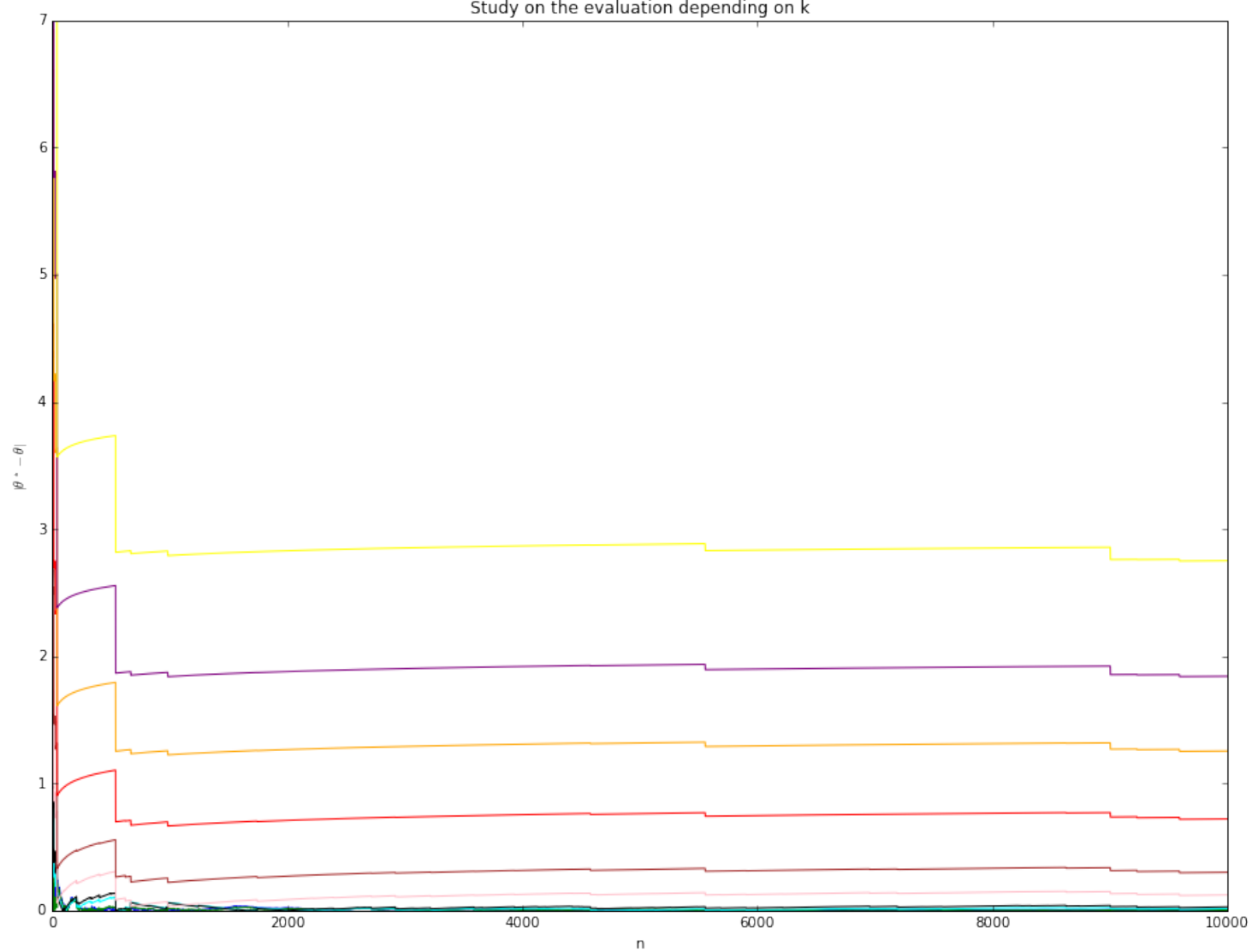
plt.subplot(211)
plt.ylim((0, 7))
plt.title('Study on the evaluation depending on k')
plt.ylabel(r'$|\theta^* - \theta|$')
plt.xlabel('n')

plt.subplot(212)
plt.xlabel('n')
plt.ylabel(r'$|\theta^* - \theta|$')
plt.title('Zoomed graph')
plt.ylim((0, 0.5))

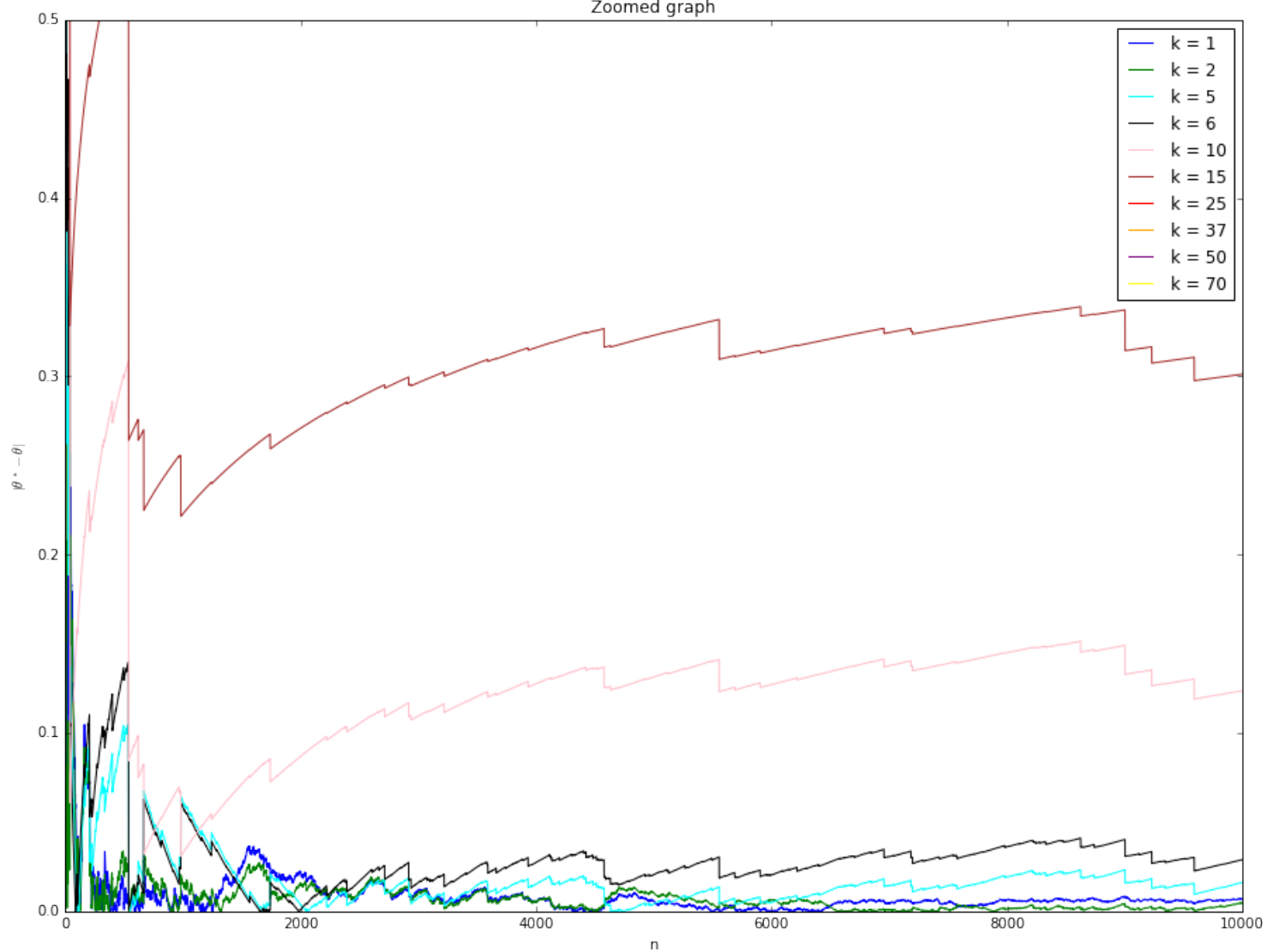
plt.show()

print('Наилучшая оценка при k =', mink)
```

Study on the evaluation depending on k



Zoomed graph



Наилучшая оценка при  $k = 2$

Исходя из графиков, можно заключить, что, в целом, с ростом  $k$  качество оценки значительно ухудшается