Задача 4.2

In [125]:

```
%matplotlib inline
import scipy.stats as st
import numpy as np
import matplotlib.pyplot as plt
import math
import statsmodels.api as sm
import csv
```

In [126]:

```
N = 1000
m = 50
sigma = 2.1
par1 = st.beta.rvs(4, 4, loc=0, scale=1, size=1)
sample1 = st.binom.rvs(m, par1, loc=0, size=N)
sample2 = st.expon.rvs(loc=0, scale=par1, size=N)
sample3 = st.norm.rvs(loc=par1, scale=math.sqrt(sigma), size=N)
```

In [127]:

```
stat = np.zeros((3, N+1))
for n in range(1, N+1):
    stat[0][n] = np.mean(sample1[:n]) / m
    stat[1][n] = np.mean(sample2[:n])
    stat[2][n] = np.mean(sample3[:n])
```

Найдем бутстреповскую оценку дисперсии для данных оценок параметров

In [128]:

```
Par = 500

samples = np.zeros(3)
disp_boot_stat1 = np.zeros((3, 1001))
ix = np.zeros((3, 1001))
for i in range(1,N+1):
    boot_stat1 = [[] for i in range(3)]
    for j in range(Par):
        samples = st.binom.rvs(m, size=i, p=stat[0][i])
        boot_stat1[0].append(np.mean(samples) / m)
        samples = st.expon.rvs(size = i, loc=0, scale=(stat[1][i]))
        boot_stat1[1].append(np.mean(samples))
        samples = st.norm.rvs(size = i, loc = stat[2][i], scale=np.sqrt(2.1))
        boot_stat1[2].append(np.mean(samples))
    for q in range (3):
        disp_boot_stat1[q][i] = np.var(np.array(boot_stat1[q]))
```

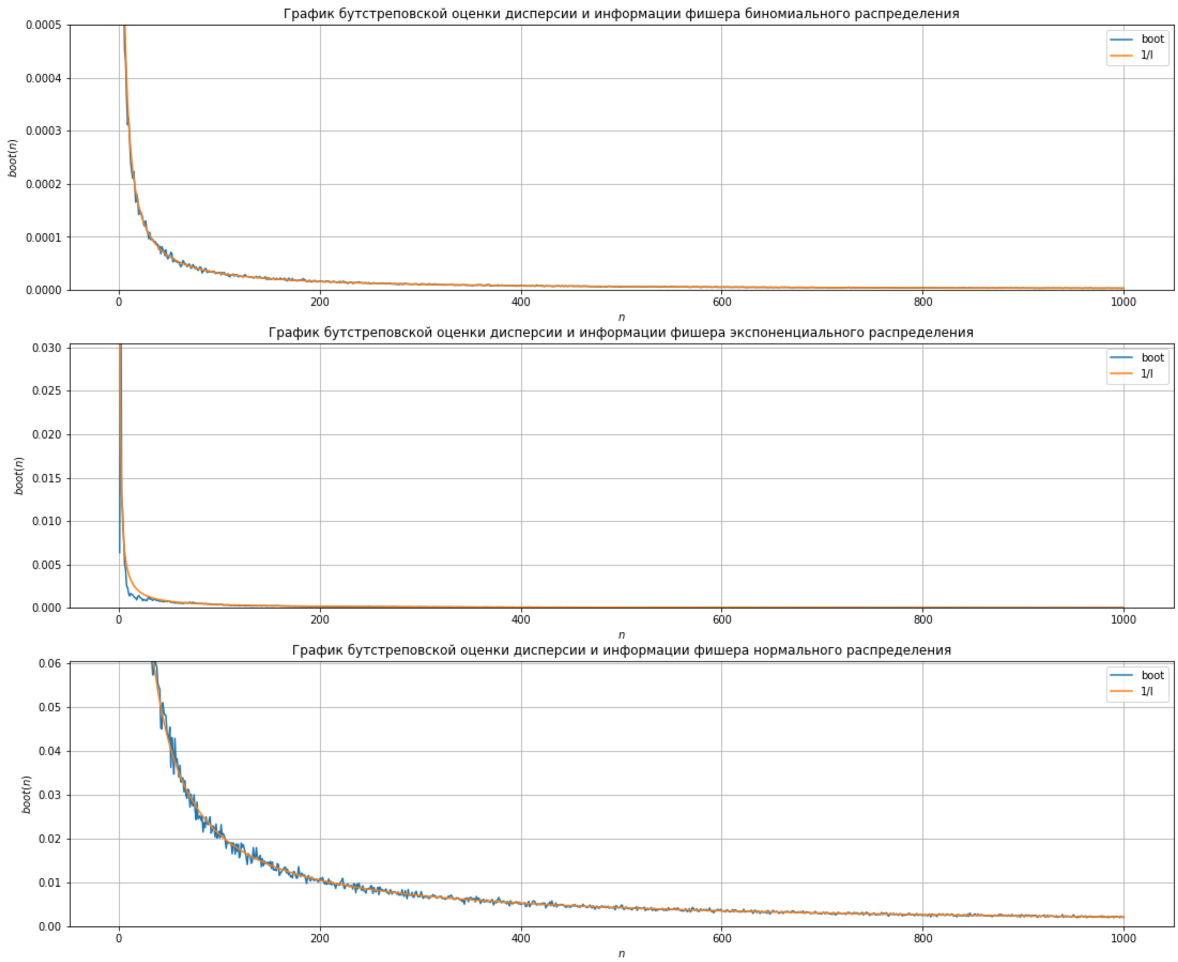
Введем функции для расчета информации Фишера

In [129]:

```
def bin_func(x, par1, m):
    return (par1 * (1 - par1) / m) / x
def exp_func(x, par1):
    return (par1**2) / x
def norm_func(x, sigma):
    return sigma/x
```

In [130]:

```
titles = ["биномиального распределения", "экспоненциального распределения", "нормального ра
for i in range(3):
    x = np.arange(N+1) + 1
    plt.subplot(3, 1, i+1)
    plt.plot(np.arange(1,1001), disp_boot_stat1[i][1:], label='boot')
    if i==0:
        plt.plot(np.arange(1,1001), bin_func(np.arange(1,1001), par1, m), label = '1/I')
    if i==1:
        plt.plot(np.arange(1,1001), exp_func(np.arange(1,1001), par1), label = '1/I')
    if i==2:
        plt.plot(np.arange(1,1001), norm_func(np.arange(1,1001), sigma), label = '1/I')
    plt.title(r'График бутстреповской оценки дисперсии и информации фишера '+titles[i]) #За
    plt.xlabel(r'$n$') #Метка по оси x в формате TeX
    plt.ylabel(r'$boot(n)$') #Метка по оси y в формате TeX
    plt.legend()
    plt.ylim(0, 0.0005+i*0.03)
    plt.grid(True) #Сетка
    fig = plt.gcf()
    fig.set_size_inches(18.5, 15.5)
```



Нижняя граница дисперсии ошибки равна $\frac{\lambda^2}{n}$, так как информация Фишера одного измерения равна $I = \frac{1}{\lambda^2}$. Вместе с тем анализ всех возможных оценок показывает, что нижней границы достичь не удается. Минимальную дисперсию $\frac{\lambda^2}{n-2}$, но большую чем I^{-1}, имеет эффективная несмещенная оценка $\lambda_0 = \frac{n-1}{n\overline{X}}$. Изменим условия этого примера и поставим задачу оценки параметра $\theta = \frac{1}{\lambda}$ экспоненциального распределения. И в качестве оценки для биномального распредления возьмем $\frac{X_1}{m}$, а для нормального распределения выборочную медиану.

In [131]:

```
N = 1000
m = 50
sigma = 2.1
par1 = st.beta.rvs(4, 4, loc=0, scale=1, size=1)
sample1 = st.binom.rvs(m, par1, loc=0, size=N)
sample2 = st.expon.rvs(loc=0, scale=par1, size=N)
sample3 = st.norm.rvs(loc=par1, scale=math.sqrt(sigma), size=N)
```

In [132]:

```
stat = np.zeros((3, N+1))
for n in range(1, N+1):
    stat[0][n] = np.min(sample1[:n]) / m
    stat[1][n] = (n+1) * np.mean(sample2[:n+1]) / (n)
    stat[2][n] = np.median(sample3[:n])
```

Аналогично найдем бутстреповскую оценку дисперсии уже для других оценок параметров

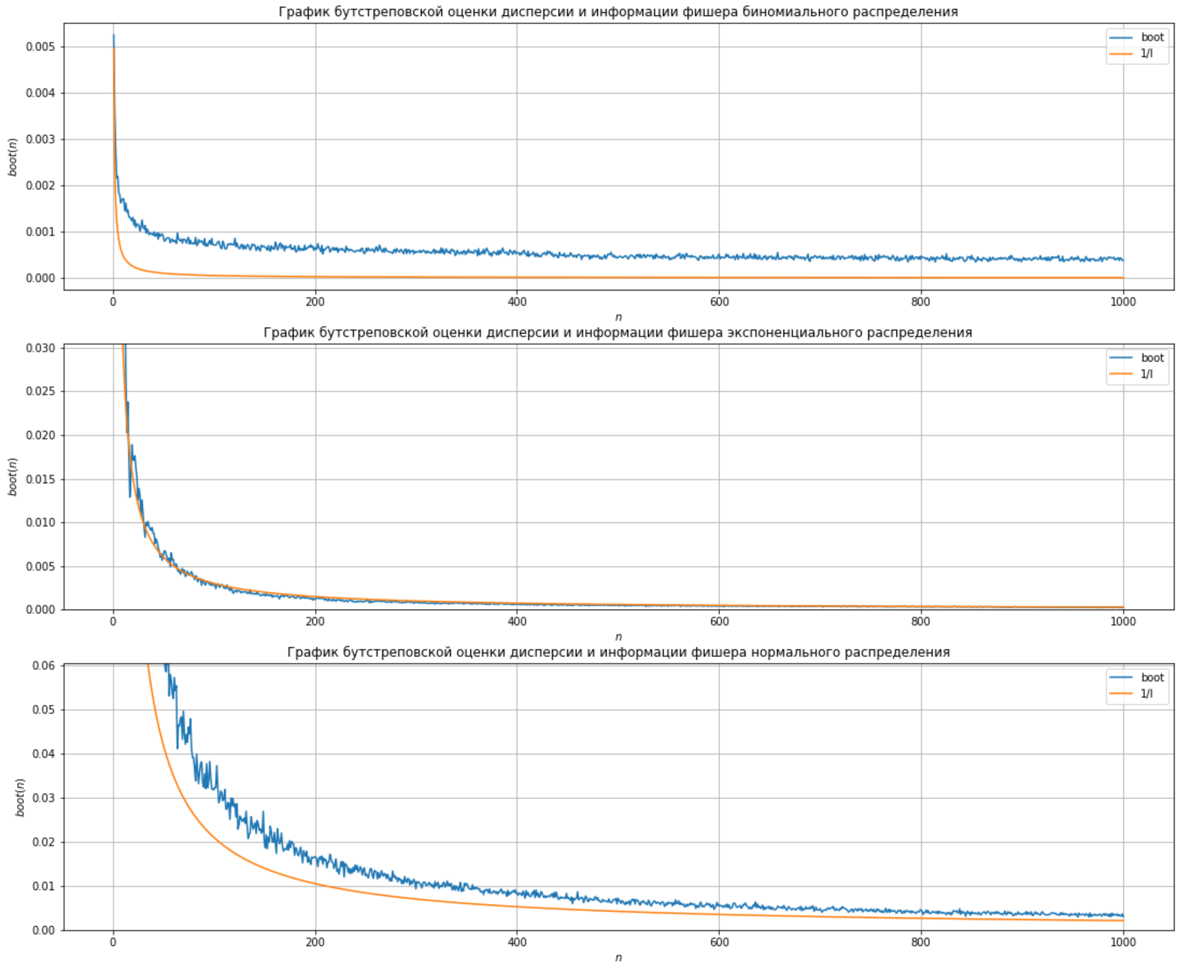
In [133]:

```
Par = 500

samples = np.zeros(3)
disp_boot_stat2 = np.zeros((3, 1001))
ix = np.zeros((3, 1001))
for i in range(1,N+1):
    boot_stat1 = [[] for i in range(3)]
    for j in range(Par):
        samples = st.binom.rvs(m, size=i, p=stat[0][i])
        boot_stat1[0].append(np.min(samples[:i]) / m)
        samples = st.expon.rvs(size = i, loc=0, scale=(stat[1][i]))
        boot_stat1[1].append((i+1) * np.mean(samples[:i+1]) / (i))
        samples = st.norm.rvs(size = i, loc = stat[2][i], scale=np.sqrt(2.1))
        boot_stat1[2].append(np.median(samples[:i]))
    for q in range (3):
        disp_boot_stat2[q][i] = np.var(np.array(boot_stat1[q]))
```

In [135]:

```
titles = ["биномиального распределения", "экспоненциального распределения", "нормального ра
for i in range(3):
    x = np.arange(N+1) + 1
    plt.subplot(3, 1, i+1)
    plt.plot(np.arange(1,1001), disp_boot_stat2[i][1:], label='boot')
    if i==0:
        plt.plot(np.arange(1,1001), bin_func(np.arange(1,1001), par1, m), label = '1/I')
    if i==1:
        plt.plot(np.arange(1,1001), exp_func(np.arange(1,1001), par1), label = '1/I')
    if i==2:
        plt.plot(np.arange(1,1001), norm_func(np.arange(1,1001), sigma), label = '1/I')
    plt.title(r'График бутстреповской оценки дисперсии и информации фишера '+titles[i]) #За
    plt.xlabel(r'$n$') #Метка по оси x в формате TeX
    plt.ylabel(r'$boot(n)$') #Метка по оси y в формате TeX
    plt.legend()
    if i != 0:
        plt.ylim(0, 0.0005+i*0.03)
    plt.grid(True) #Сетка
    fig = plt.gcf()
    fig.set_size_inches(18.5, 15.5)
```



Вывод

Сформилируем неравенство Рао-Крамера

Неравенством Рао-Крамера называется неравенство, которое при некоторых условиях на статистическую модель даёт нижнюю границу для дисперсии оценки неизвестного параметра, выражая её через информацию Фишера. Часто используется следующий частный случай вышеприведённого неравенства. Пусть выполнены условия регулярности, а $\widehat{\theta}(x)$ — несмещённая оценка параметра θ. Тогда : $\mathrm{D}_\theta\, \widehat{\theta}(x) \geqslant \frac{1}{I_n(\theta)}$. Равенство в этом неравенстве достигается тогда и только тогда, когда $\hat{\theta}(x) - \theta = a(\theta)U(\theta, x)$

Заметим, что на эффективных оценках(первые 3 графика) бутстрепная оценка дисперсии приближает нижнюю оценку дисперсии из неравенства Рао-Крамера. А в последних 3 графиках видно, что неравенство Рао-Крамера невыполняется на экспоненциальном распределении, так как оценка не является несмещенной. В остальных распределениях график бутстреповской оценки дисперсии лежит выше графика информации Фишера, что и иллюстрирует неравенство Рао-Крамера.