Статистика, прикладной поток

Практическое задание 2

В данном задании вы визуализируйте некоторые свойства оценок (несмещенность, состоятельность, асимптотическая нормальность), посмотрите на свойства оценки максимального правдоподобия, а также сравните некоторые оценки при помощи построения функций риска.

Правила:

- Дедлайн **13 октября 23:59**. После дедлайна работы не принимаются кроме случаев наличия уважительной причины.
- Выполненную работу нужно отправить на почту `mipt.stats@yandex.ru`, указав тему письма "`[applied] Фамилия Имя - задание 2`". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Прислать нужно ноутбук и его pdf-версию (без архивов). Названия файлов должны быть такими: `2.N.ipynb` и `2.N.pdf`, где `N` - ваш номер из таблицы с оценками.
- Решения, размещенные на каких-либо интернет-ресурсах не принимаются. Кроме того, публикация решения в открытом доступе может быть приравнена к предоставлении возможности списать.
- Для выполнения задания используйте этот ноутбук в качествие основы, ничего не удаляя из него.
- Никакой код из данного задания при проверке запускаться не будет.

Баллы за задание:

- Задача 1 - 10 баллов
- Задача 2 - 5 баллов
- Задача 3 - 5 баллов
- Задача 4 - 5 баллов
- Задача 5 - 5 баллов
- Задача 6 - 20 баллов

Все задачи имеют тип **О2**. Подробнее см. в правилах выставления оценки.

In [3]:

```
import numpy as np
import scipy.stats as sps
from scipy.special import factorial
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

Задача 1. В этой задаче предлагается изучить *свойство несмещённости*.

1. Пусть $X_1, \ldots, X_n$ --- выборка из распределения $U[0, \theta]$. Рассмотрим оценки $X_{(n)}, \frac{n+1}{n} X_{(n)}, 2\overline{X}$ параметра θ.

Какие из этих оценок являются несмещенными?

Ответ: Оценки $\frac{n+1}{n} X_{(n)}, 2\overline{X}$ являются несмещенными, оценка $X_{(n)}$ смещенная

Теперь проверьте это на практике. Для каждой из приведенных выше оценок $\hat{\theta}$:

Вычислите $k = 500$ независимых оценок $\hat{\theta}_1, \ldots, \hat{\theta}_k$ по независимым выборкам $(X_1^1, \ldots, X_n^1), \ldots, (X_1^k, \ldots, X_n^k)$, сгенерированным из распределения $U[0, 1]$. Далее вычислите среднее этих оценок, которое обозначим $\overline{\theta}$.

Визуализируйте полученные значения, построив на **одном** графике точки $(\hat{\theta}_1$, y$), \ldots, (\hat{\theta}_k, y)$ и среднее оценок $(\overline{\theta}, y)$, где y -- произвольные различные (например 0, 1, 2) координаты для трёх различных типов оценок.

Повторите действие три раза для $n \in \{10, 100, 500\}$. В итоге получится три графика для различных n, на каждом из которых изображено поведение трёх типов оценок и их среднее.

Копипаста неприемлема, используйте циклы и функции.

Используйте данный шаблон для визуализации значений:

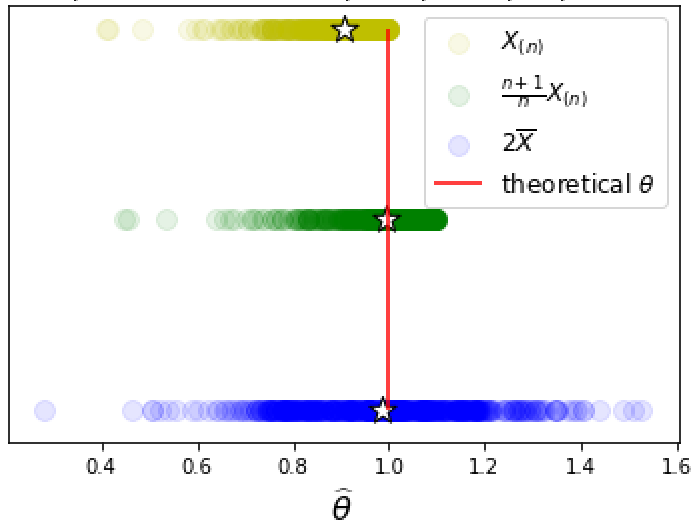
In [21]:

```
def calculate_estimators(sample):
    maximum = np.max(sample, axis=0)
    average = np.average(sample, axis=0)
    n = sample.shape[0]

    return maximum, maximum*(n+1)/n, average*2

def show_theta_estimators(sample, theta=1):
    theta_estimators = np.array(calculate_estimators(sample))

    estimator_labels = ['$X_{(n)}$',
                        "$\\frac{n+1}{n}X_{(n)}$",
                        '$2\overline{X}$']

    colors = ['y', 'g', 'b']

    plt.figure()
    # Для каждой оценки:
    for i in range(3):
        plt.scatter(theta_estimators[i] , np.zeros(sample.shape[1]) + 2-i,
                    alpha=0.1, s=100, color=colors[i], label=estimator_labels[i])
        plt.scatter(theta_estimators[i].mean(), 2-i, marker='*', s=200,
                    color='w', edgecolors='black')

    # Для всего графика:
    plt.vlines(1, 0, 2, color='r', label='theoretical $\\theta$')
    plt.title('Распределение оценок $\\theta$, размер выборок равен %d' % sample.shape[
0])
    plt.yticks([])
    plt.xlabel("$\\widehat\\theta$", fontsize=16)
    plt.legend(fontsize=12)
    plt.show()
```

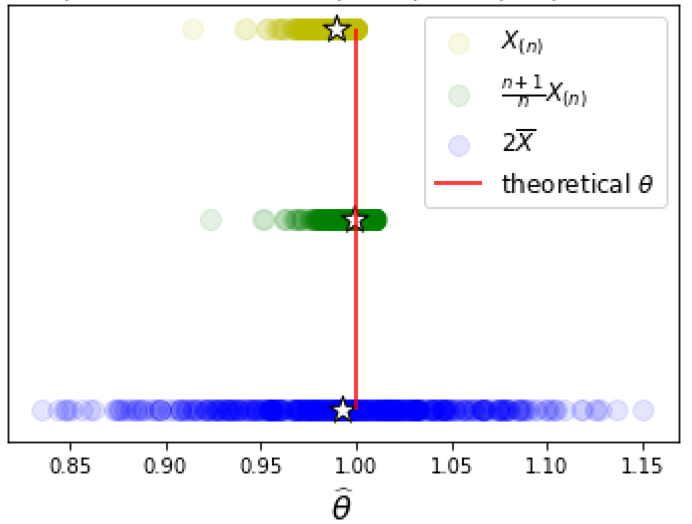
In [22]:

```
sample = sps.uniform.rvs(size=(500, 500))
for n in [10, 100, 500]:
    show_theta_estimators(sample=sample[:n, :], theta=1)
```

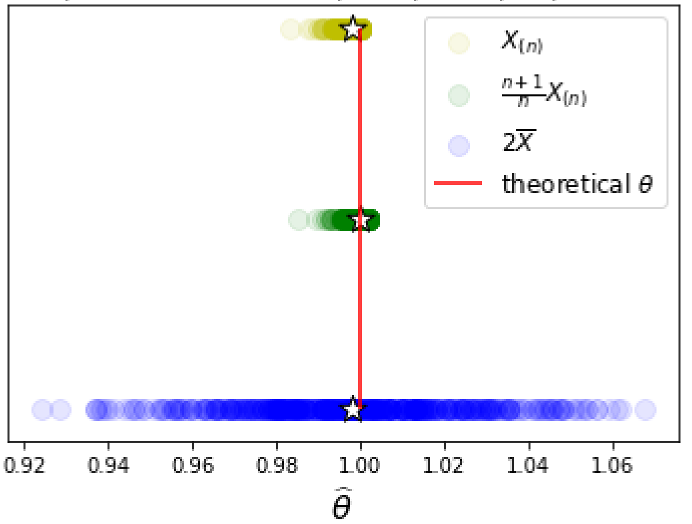
Распределение оценок θ, размер выборок равен 10



Распределение оценок θ, размер выборок равен 100



Распределение оценок θ, размер выборок равен 500



Вывод: При малых n можно заметить, что оценка $X_{(n)}$ смещенная, но при больших это становится менее заметно, потому что $\frac{n+1}{n} \to 1$

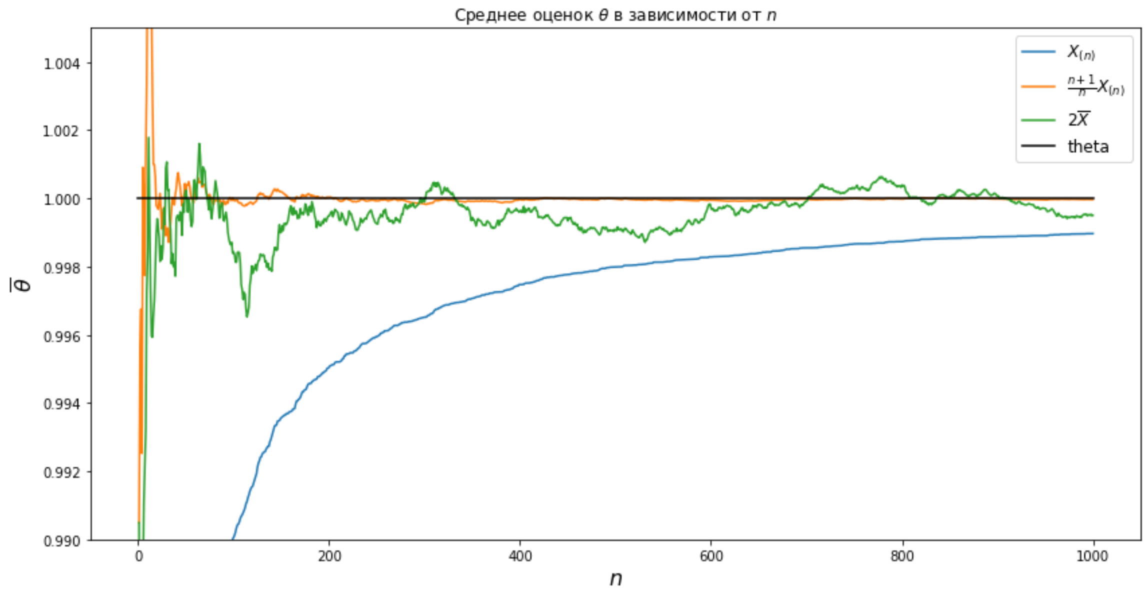
---

2. Изучим поведение среднего оценок из первого пункта при росте размера n выборки. Постройте график зависимости $\bar{\theta}$ от n для трёх типов оценок. Какие из оценок являются асимптотически несмещёнными (т.е. $\forall \theta \in \Theta: \mathrm{E}_\theta \hat{\theta} \to \theta$ при $n \to +\infty$)?

Ответ: все три оценки асимптотически несмещённые

In [23]:

```
def draw_estimator_from_n(theta=1, k=500, max_n=1000):
    sample = sps.uniform.rvs(size=(max_n, k), loc=0, scale=theta)

    mean_cummax = np.maximum.accumulate(sample, axis=0).mean(axis=1)
    mean_theta_estinaions = np.array([mean_cummax,
                                     mean_cummax*np.arange(2, max_n+2)/np.arange(1, max_n+1
),
                                     np.cumsum(sample, axis=0).mean(axis=1)*2/np.arange(1,
max_n+1)])

    estimator_labels = ['$X_{(n)}$',
                        '$\\frac{n+1}{n}X_{(n)}$',
                        '$2\overline{X}$']


    plt.figure(figsize=(14, 7))
    # Для каждой оценки:
    for i, theta_estimator in enumerate(mean_theta_estinaions):
        plt.plot(np.linspace(1, max_n, max_n), theta_estimator,
                 label=estimator_labels[i])

    # Для всего графика:
    plt.plot([0, max_n], [theta, theta], label='theta', color='black')
    plt.title('Среднее оценок $\\theta$ в зависимости от $n$')
    plt.ylim(theta-0.01, theta+0.005)
    plt.xlabel('$n$', fontsize=16)
    plt.ylabel('$\\overline{\\theta}$', fontsize=16)
    plt.legend(fontsize=12)
    plt.show()

draw_estimator_from_n()
```



Вывод: Из графика видно, что все три оценки асимптотически несмещенные, так как графики оценок стремятся к истиному значению θ

---

3. Пусть теперь $X_1, \ldots, X_n$ --- выборка из распределения $\mathcal{N}(0, \sigma^2)$. Известно, что в качестве оценки параметра σ^2 можно использовать следующие оценки S^2, $\frac{n}{n-1}S^2$. Какие из этих оценок являются несмещенными?

Напоминание: $S^2 = \frac{1}{n}\sum\limits_{i=1}^{n}(X_i - \overline{X})^2 = \overline{X^2} - \overline{X}^2$

Ответ: S^2 смещенная оценка, $\frac{n}{n-1}S^2$ несмещенная. Обе асимтотически несмещенные

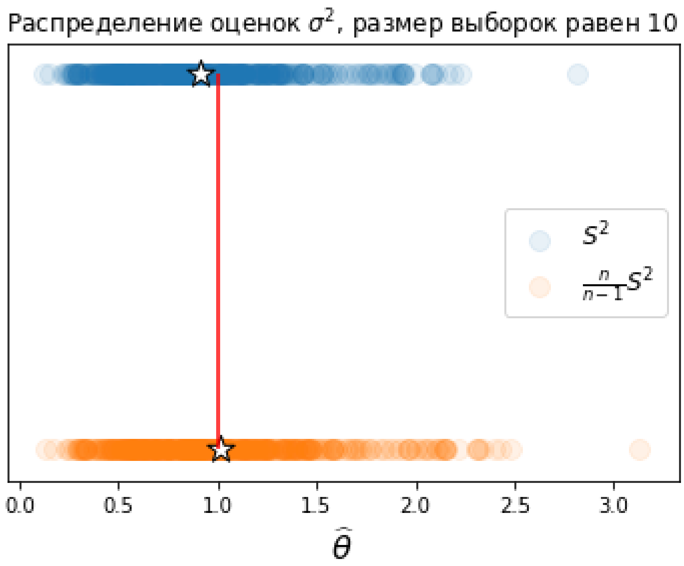
Для данной модели повторите действия из первых двух частей.

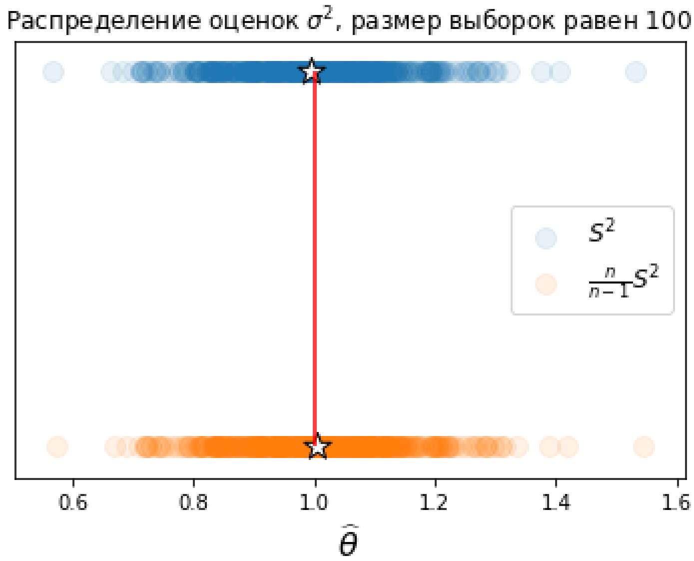
In [30]:

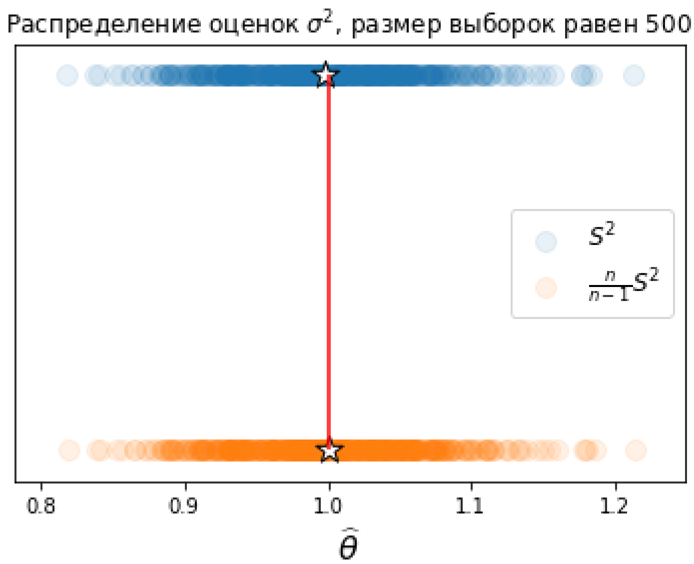
```
def calculeate_sigma_estimators(sample):
    return np.var(sample, axis=0), np.var(sample, axis=0, ddof=1)

def draw_sigma_estimators(sample, theta=1):
    sigma_estimators = np.array(calculeate_sigma_estimators(sample))

    estimator_labels = ['$S^2$',
                        '$\\frac{n}{n-1}S^2$']

    plt.figure()
    # Для каждой оценки:
    for i in range(2):
        plt.scatter(sigma_estimators[i] , np.zeros(sample.shape[1]) + 1-i,
                  alpha=0.1, s=100, label=estimator_labels[i])

        plt.scatter(sigma_estimators[i].mean(), 1-i, marker='*', s=200,
                 color='w', edgecolors='black')

    # Для всего графика:
    plt.vlines(1, 0, 1, color='r')
    plt.title('Распределение оценок $\\sigma^2$, размер выборок равен %d' % sample.shap
e[0])
    plt.yticks([])
    plt.xlabel("$\\widehat\\theta$", fontsize=16)
    plt.legend(fontsize=12)
    plt.show()
```

In [31]:

```
sample = sps.norm.rvs(size=(500, 500))
for n in [10, 100, 500]:
    draw_sigma_estimators(sample=sample[:n, :], theta=1)
```


Распределение оценок σ^2, размер выборок равен 10
S^2
$\frac{n}{n-1}S^2$
0.0
0.5
1.0
1.5
2.0
2.5
3.0
$\widehat{\theta}$

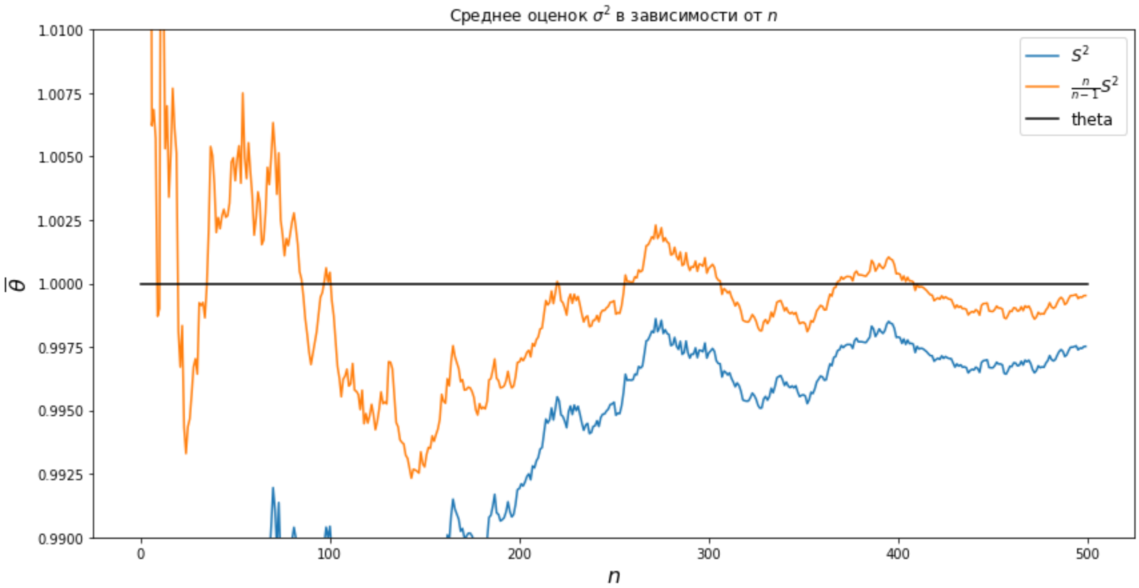

Распределение оценок σ^2, размер выборок равен 100
S^2
$\frac{n}{n-1}S^2$
0.6
0.8
1.0
1.2
1.4
1.6
$\widehat{\theta}$


Распределение оценок σ^2, размер выборок равен 500
S^2
$\frac{n}{n-1}S^2$
0.8
0.9
1.0
1.1
1.2
$\widehat{\theta}$

Из графиков видно, что при $n = 10$ оценка S^2 существенно смещена, при больших значениях n смещение становится меньше. Оценка $\frac{n}{n-1}S^2$ всегда несмещенная.

In [37]:

```
def calculeate_sigma_estimators(sample):
    n = sample.shape[0]
    mean = np.cumsum(sample, axis=0)/np.full(sample.shape, np.arange(1, n+1)).T
    second_moment= np.cumsum(sample*sample, axis=0)/np.full(sample.shape, np.arange(1,
n+1)).T
    S2 = second_moment - mean**2
    S2 = S2.mean(axis=1)
    S2=S2[2:]

    return S2, S2*np.arange(2, n)/(np.arange(1, n-1))

def draw_sigma_estimator_from_n(theta=1, k=500, max_n=500):
    sample = sps.norm.rvs(size=(max_n, k), loc=0, scale=theta)

    sigma_estimators = np.array(calculeate_sigma_estimators(sample))

    estimator_labels = ['$S^2$',
                        '$\\frac{n}{n-1}S^2$']

    plt.figure(figsize=(14, 7))
    # Для каждой оценки:
    for i in range(2):
        plt.plot(np.arange(2, max_n), sigma_estimators[i],
                 label=estimator_labels[i])

    # Для всего графика:
    plt.plot([0, max_n], [theta, theta], label='theta', color='black')
    plt.title('Среднее оценок $\\sigma^2$ в зависимости от $n$')
    plt.ylim(theta-0.01, theta+0.01)
    plt.xlabel('$n$', fontsize=16)
    plt.ylabel('$\\overline{\\theta}$', fontsize=16)
    plt.legend(fontsize=12)
    plt.show()

draw_sigma_estimator_from_n()
```



Сделайте вывод о том, что такое свойство несмещенности. Подтверждают ли сделанные эксперименты свойство несмещенности данных оценок? Поясните, почему в лабораторных по физике при оценке погрешности иногда используют $n-1$ в знаменателе, а не n.

Вывод: Оценка $\hat{\theta}$ несмещена относительно параметра θ, если при большом колличестве выборок среднее от оценок примерно равно истиному значению θ.

Асимптотическая несмещенность означает, что для достаточно больших размерах выборки оценка неотличима от несмещенной.

В лабораторных по физике делили на $n-1$, при малых n. Так делали потому что $\frac{n}{n-1}S^2$ несмещена и быстрее сходится.

Задача 2. В этой задаче нужно визуализировать *свойство состоятельности*.

Пусть $X_1, \ldots, X_n$ --- выборка из распределения $U(0, \theta)$. Из домашнего задания известно, что оценки $\theta^* = 2\overline{X}, \hat{\theta} = X_{(n)}$ являются состоятельными оценками θ. Вам нужно убедиться в этом, сгенерировав множество выборок, посчитав по каждой из них указанные выше оценки параметра θ в зависимости от размера выборки и визуализировав их состоятельность.

Сгенерируйте множество выборок $X^1, \ldots, X^{300}$ из распределения $U[0, 1]$: $X^j = (X^j_1, \ldots, X^j_{500}), 1 \leq j \leq 300$.

По каждой из них посчитайте оценки $\theta^*_{jn} = 2\frac{X^j_1 + \cdots + X^j_n}{n}$, $\hat{\theta}_{jn} = \max(X^j_1, \ldots, X^j_n)$ для $1 \leq n \leq 500$, то есть оценки параметра θ по первым n наблюдениям j-й выборки. При написании кода могут помочь функции `numpy.cumsum(axis=...)` и `np.maximum.accumulate(axis=...)`.

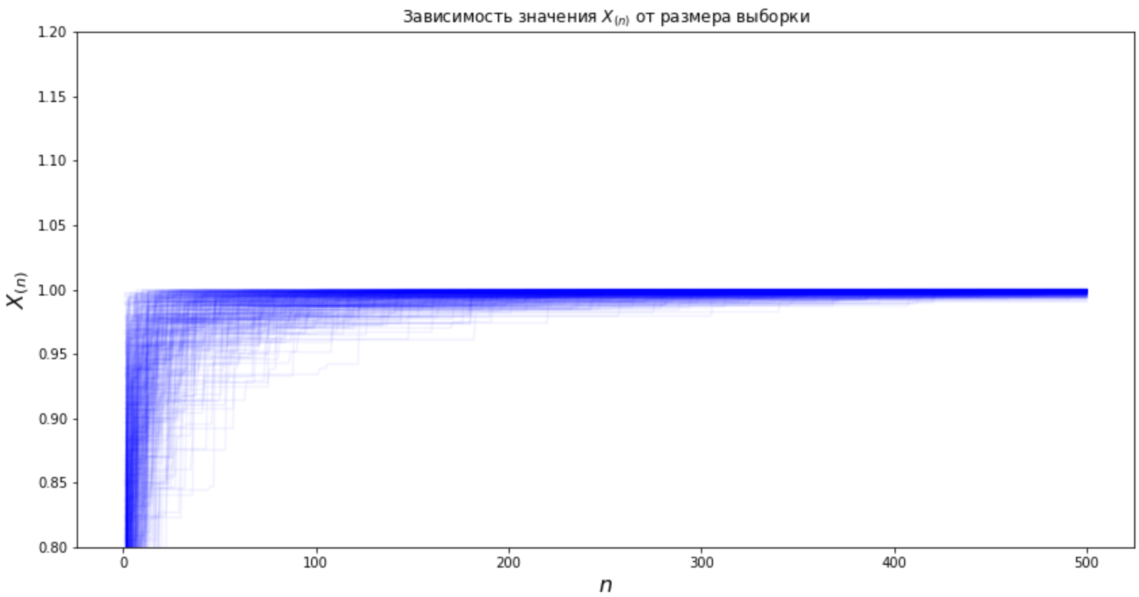
In [39]:

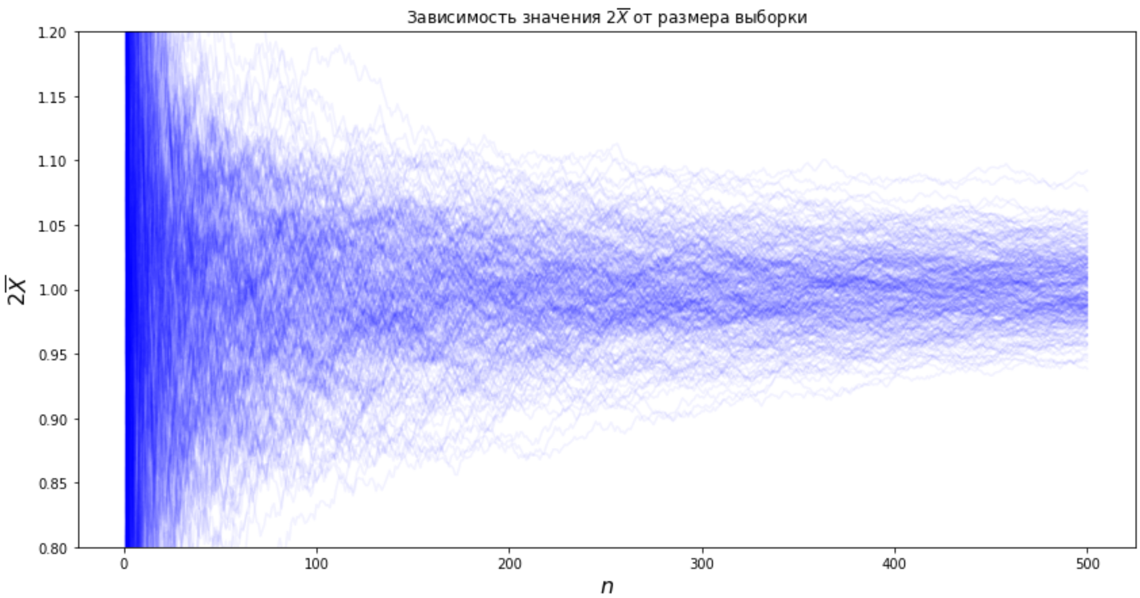
```
j, n = 300, 500
sample = sps.uniform.rvs(size=(j, n))
estimators1 = np.maximum.accumulate(sample, axis=1)
estimators2 = np.cumsum(sample, axis=1)*2/np.full((j, n), np.arange(1, n+1))
```

Для каждой оценки $\theta^*, \hat{\theta}$ нарисуйте следующий график. Для каждого j нанесите на один график зависимости θ^*_{jn} или $\hat{\theta}_{jn}$ от n с помощью `plt.plot`. Каждая кривая должна быть нарисована *одним цветом* с прозрачностью `alpha=0.05`. Поскольку при малых n значения средних могут быть большими по модулю, ограничьте область графика по оси *y* с помощью функции `plt.ylim((min, max))`.

In [52]:

```
def draw_estimators(estimators, label, ymin=1-0.2, ymax=1+0.2, alpha=0.05):
    plt.figure(figsize=(14, 7))
    for estimator in estimators:
        plt.plot(np.arange(1, n+1), estimator, alpha=alpha, color='b')

    plt.title("Зависимость значения "+label+" от размера выборки")
    plt.ylim(ymin, ymax)
    plt.xlabel('$n$', fontsize=16)
    plt.ylabel(label, fontsize=16)
    plt.show()

draw_estimators(estimators1, label='$X_{(n)}$')
draw_estimators(estimators2, label='$2\overline{X}$')
```





Сделайте вывод о смысле закона больших чисел. Подтверждают ли сделанные эксперименты теоретические свойства?

Вывод: Смысл ЗБЧ в том, что среднее от выборки случайной величины сходится к матожиданию случайной величины при росте рамера выборки. На графике можно увидеть, что все оценки постепенно сходятся к $\theta = 1$.

Задача 3. В этой задаче нужно визуализировать *свойство асимптотической нормальности*.

а). Пусть $X_1, \ldots, X_n$ --- выборка из распределения $U(0, 1)$. Согласно центральной предельной теореме оценка $\theta^* = 2\overline{X}$ является асимптотически нормальной оценкой параметра θ. Вам нужно убедиться в этом, сгенерировав множество наборов случайных величин и посчитав по каждому из наборов величину $Z_n = \sqrt{n}\left(2\overline{X} - \theta\right)$ в зависимости от размера набора.

Сгенерируйте множество выборок $X^1, \ldots, X^{300}$ из распределения $U[0, 1]$: $X^j = (X^j_1, \ldots, X^j_{500}), 1 \le j \le 300$. По каждой из них посчитайте оценки $\theta^*_{jn} = 2\frac{X^j_1 + \cdots + X^j_n}{n}$ для $1 \le n \le 500$, то есть оценку параметра θ по первым n наблюдениям j-й выборки. Для этих оценок посчитайте статистики $Z_{jn} = \sqrt{n}\left(\theta^*_{jn} - \theta\right)$, где $\theta = 1$.

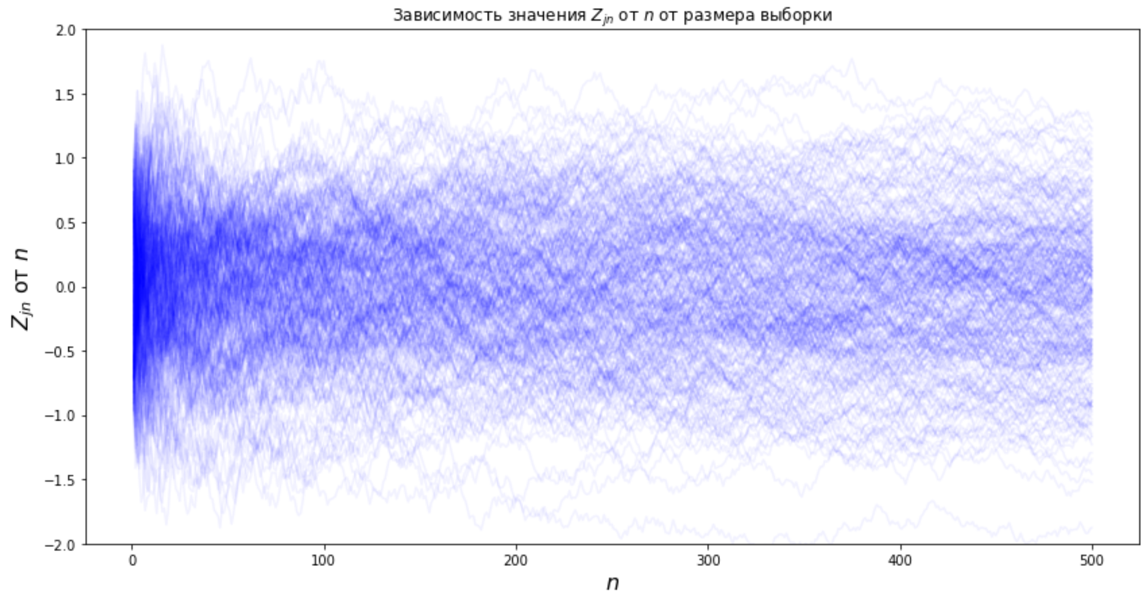
In [53]:

```
theta = 1
j, n = 300, 500
sample = sps.uniform.rvs(size=(j, n), scale=theta)
estimators = np.cumsum(sample, axis=1)*2/np.full((j, n), np.arange(1, n+1))

Z = (estimators-theta)*np.full((j, n), np.sqrt(np.arange(1, n+1)))
```

Для каждого j нанесите на один график зависимость Z_{jn} от n с помощью `plt.plot`. Каждая кривая должна быть нарисована *одним цветом* с прозрачностью `alpha=0.05`. Сходятся ли значения Z_{jn} к какой-либо константе?

In [54]:

```
draw_estimators(Z, label='$Z_{jn}$ от $n$', ymin=-2, ymax=2)
```

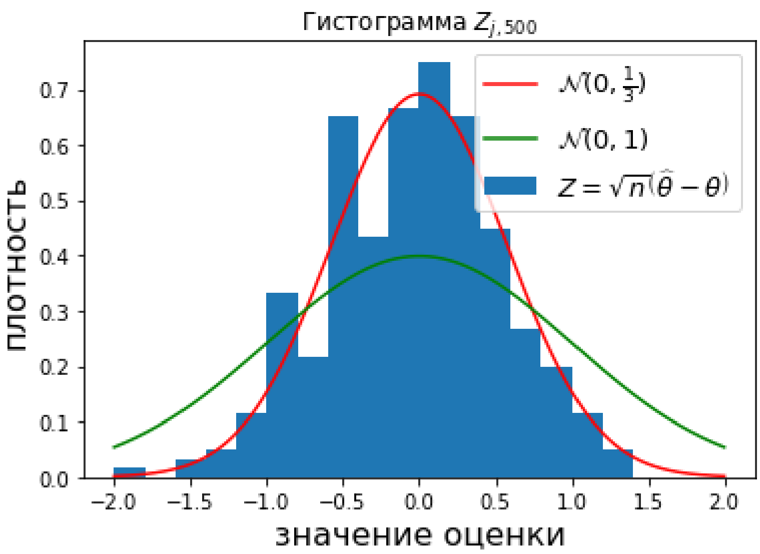


Значения Z_{jn} не сходятся к константе

Для $n = 500$ по выборке $Z_{1,500}, \ldots, Z_{300,500}$ постройте гистограмму и график плотности распределения $\mathcal{N}(0, 1)$. Не забудьте сделать легенду.

In [64]:

```
grid = np.linspace(-2, 2, 1000)
plt.figure()
plt.plot(grid, sps.norm.pdf(grid, scale=1/3**0.5), color='red', label='$\\mathcal{N}(0,
 \\frac{1}{3})$')
plt.plot(grid, sps.norm.pdf(grid, scale=1), color='g', label='$\\mathcal{N}(0, 1)$')
plt.hist(Z[:, -1], range=(-2, 2), bins=20, density=True, label='$Z = \\sqrt{n} \\left(
\\widehat\\theta - \\theta \\right)$')
plt.legend(fontsize=13)
plt.xlabel('значение оценки', fontsize=16)
plt.ylabel('плотность', fontsize=16)
plt.title('Гистограмма $Z_{j,500}$')
plt.show()
```



Сделайте вывод о смысле свойства асимптотической нормальности. Подтверждают ли сделанные эксперименты теоретические свойства?

Ассимптотическая нормальность оценки $\hat{\theta}$ означает, что $\sqrt{n}\left(\hat{\theta} - \theta\right) \xrightarrow{d_\theta} \mathcal{N}(0, \sigma^2(\theta))$, где $\sigma^2(\theta)$ - асимптотическая дисперсия. Из графика видно, что $\sqrt{n}\left(\hat{\theta} - \theta\right) \xrightarrow{d_\theta} \mathcal{N}(0, \frac{1}{3}))$. Значит, оценка $\theta^* = 2\overline{X}$ асимптотичеки нормальна с асимптотической дисперсией $\frac{1}{3}$

Задача 4. Пусть $X_1, \ldots, X_n$ --- выборка из распределения $U[0, \theta]$. Из домашнего задания известно, что $n\left(\theta - X_{(n)}\right) \xrightarrow{d_\theta} Exp(1/\theta)$. Вам нужно убедиться в этом, сгенерировав множество выборок, посчитав по каждой из них оценку $X_{(n)}$ параметра θ в зависимости от размера выборки и визуализировав рассматриваемое свойство.

Сгенерируйте множество выборок $X^1, \ldots, X^{300}$ из распределения $U[0, 1]$: $X^j = (X_1^j, \ldots, X_{500}^j)$, $1 \le j \le 300$. По каждой из них посчитайте оценки $\hat{\theta}_{jn} = \max(X_1^j, \ldots, X_n^j)$ для $1 \le n \le 500$, то есть оценку параметра θ по первым n наблюдениям j-й выборки. Для этих оценок посчитайте статистики $T_{jn} = n\left(\theta - \hat{\theta}_{jn}\right)$, где $\theta = 1$.

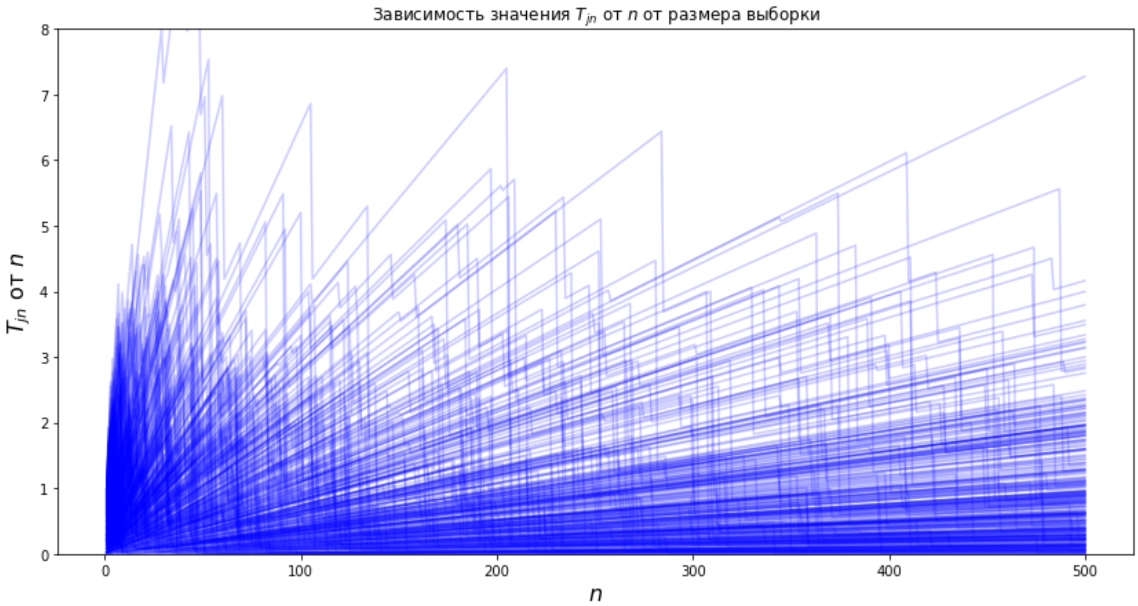
In [56]:

```
j, n = 300, 500
theta = 1
sample = sps.uniform.rvs(size=(j, n), scale=theta)
estimators = np.maximum.accumulate(sample, axis=1)
T = (theta-estimators)*np.full((j, n), np.arange(1, n+1))
```

Для каждого j нанесите на один график зависимость T_{jn} от n с помощью `plt.plot`. Все кривые должны быть нарисованы *одним и тем же цветом* с прозрачностью `alpha=0.2`. Сходятся ли значения T_{jn} к какой-либо константе?

In [57]:

```
draw_estimators(T, label='$T_{jn}$ от $n$', ymin=0, ymax=8, alpha=0.2)
```

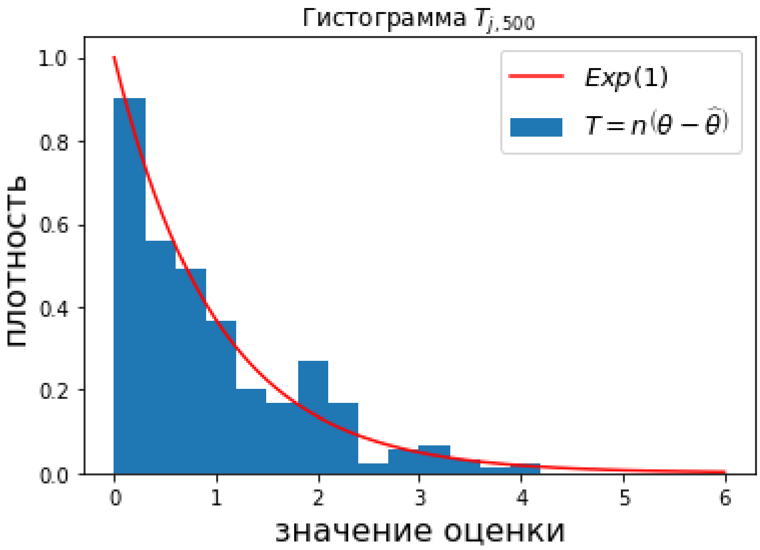


Ответ: Значения T_{jn} не сходятся к константе.

Для $n = 500$ по выборке $T_{1,500}, \ldots, T_{300,500}$ постройте гистограмму и график плотности распределения $Exp(1)$. Не забудьте сделать легенду.

In [63]:

```
grid = np.linspace(0, 6, 1000)
plt.figure()
plt.plot(grid, sps.expon.pdf(grid, scale=1), color='red', label='$Exp(1)$')
plt.hist(T[:, -1], range=(0, 6), bins=20, density=True, label='$T = n \\left( \\theta - \\widehat\\theta \\right)$')
plt.legend(fontsize=13)
plt.xlabel('значение оценки', fontsize=16)
plt.ylabel('плотность', fontsize=16)
plt.title('Гистограмма $T_{j,500}$')
plt.show()
```



Хорошо ли гистограмма приближает плотность распределения $Exp(1)$? Подтверждают ли проведенные эксперименты свойство $n\left(\theta - X_{(n)}\right) \xrightarrow{d_\theta} Exp(1/\theta)$? Что можно сказать в сравнении с оценкой, рассмотренной в предыдущей задаче?

Вывод: Гистотрамма хорошо приближает график плотности $Exp(1)$, следовательно, проведенные эксперименты подтверждают свойство $n\left(\theta - X_{(n)}\right) \xrightarrow{d_\theta} Exp(1/\theta)$. Оценка $X_{(n)}$ лучше, хотя она и не является асимптотически нормальной, так как она сходится быстрее.

Задача 5. Дана параметрическая модель и 3 выборки, состоящие из 2-3 наблюдений. Для удобства, выборки представлены в виде python-кода — каждая выборка записана как список ее элементов; множество выборок представлено как список списков, соответствующих выборкам из множества. Нужно для каждой выборки построить график функции правдоподобия.

a). Параметрическая модель $\mathcal{N}(\theta, 1)$, выборки: `[[-1, 1], [-5, 5], [-1, 5]]`

b). Параметрическая модель $Exp(\theta)$, выборки: `[[1, 2], [0.1, 1], [1, 10]]`

c). Параметрическая модель $U[0, \theta]$, выборки: `[[0.2, 0.8], [0.5, 1], [0.5, 1.3]]`

d). Параметрическая модель $Bin(5, \theta)$, выборки: `[[0, 1], [5, 5], [0, 5]]`

e). Параметрическая модель $Pois(\theta)$, выборки: `[[0, 1], [0, 10], [5, 10]]`

f). Параметрическая модель $Cauchy(\theta)$, где θ — параметр сдвига, выборки: `[[-0.5, 0.5], [-2, 2], [-4, 0, 4]]`

Выполнить задание, не создавая много кода, поможет следующая функция.

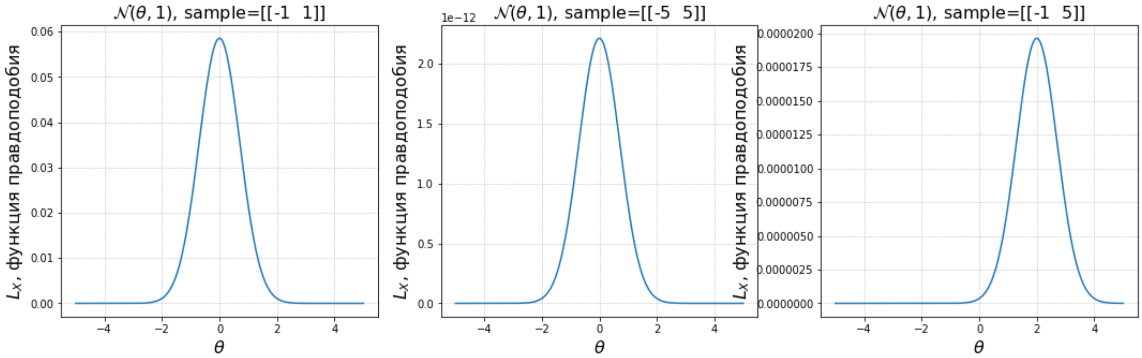
In [39]:

```
def draw_likelihood(density_function, grid, samples, label):
    """Изображает график функции правдоподобия для каждой из 3 выборок.

    Аргументы:
        density_function --- функция, считающая плотность
            (обычную или дискретную). На вход данная функция
            должна принимать массив размера (1, len_sample)
            и возвращать массив размера (len_grid, len_sample).
        grid --- массив размера (len_grid, 1), являющийся
                 сеткой для построения графика;
        samples --- три выборки;
        label --- latex-код параметрической модели.
    """

    assert len(samples) == 3, "Число выборок не равно 3."

    plt.figure(figsize=(18, 5))
    for i, sample in enumerate(samples):
        sample = np.array(sample)[np.newaxis, :]
        likelihood = np.prod(density_function(sample), axis=1)
        plt.subplot(1, 3, i+1)
        plt.plot(grid, likelihood)
        plt.xlabel('$\\theta$', fontsize=16)
        plt.ylabel('$L_X$, функция правдоподобия', fontsize=16)
        plt.grid(ls=':')
        plt.title(label + ', sample=' + str(sample), fontsize=16)
    plt.show()
```

Первый пункт можно выполнить с помощью следующего кода:

In [40]:

```
grid = np.linspace(-5, 5, 1000).reshape((-1, 1))
draw_likelihood(sps.norm(loc=grid).pdf, grid,
                [[-1, 1], [-5, 5], [-1, 5]], '$\\mathcal{N}(\\theta, 1)$')
```

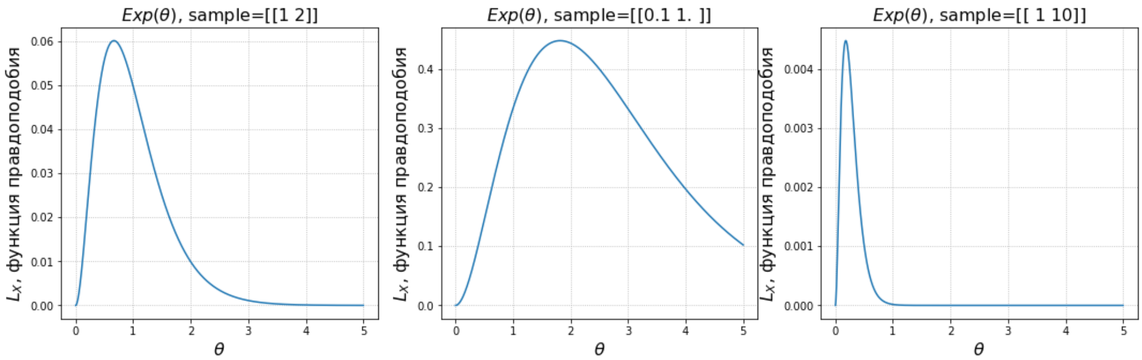


Выполните остальные:

b). Параметрическая модель $Exp(\theta)$, выборки: `[[1, 2], [0.1, 1], [1, 10]]`

In [41]:

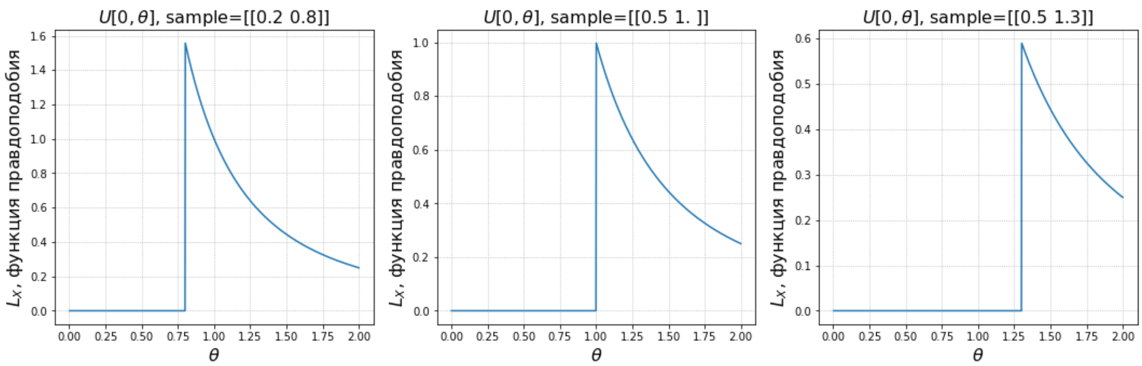
```
grid = np.linspace(0.0001, 5, 1000).reshape((-1, 1))
draw_likelihood(sps.expon(scale=1/grid).pdf, grid,
                [[1, 2], [0.1, 1], [1, 10]], '$Exp(\\theta)$')
```



c). Параметрическая модель $U[0, \theta]$, выборки: `[[0.2, 0.8], [0.5, 1], [0.5, 1.3]]`

In [42]:

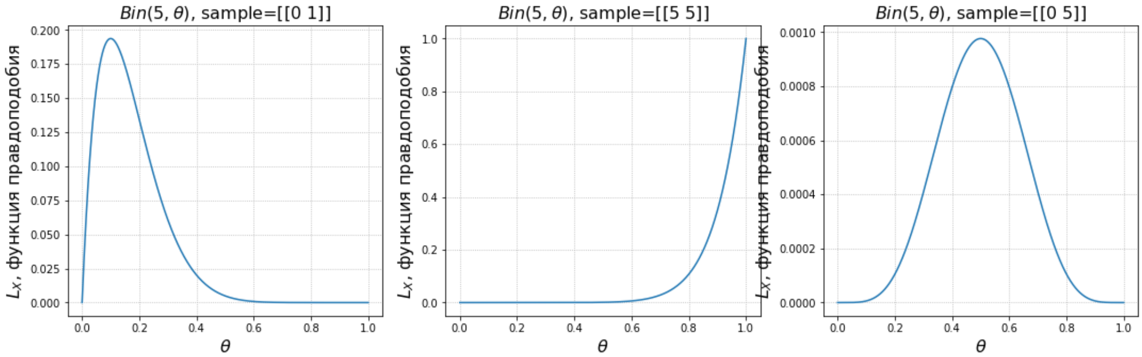
```
grid = np.linspace(0.001, 2, 1000).reshape((-1, 1))
draw_likelihood(sps.uniform(scale=grid).pdf, grid,
                [[0.2, 0.8], [0.5, 1], [0.5, 1.3]], '$U[0, \\theta]$')
```



d). Параметрическая модель $Bin(5, \theta)$, выборки: `[[0, 1], [5, 5], [0, 5]]`

In [43]:

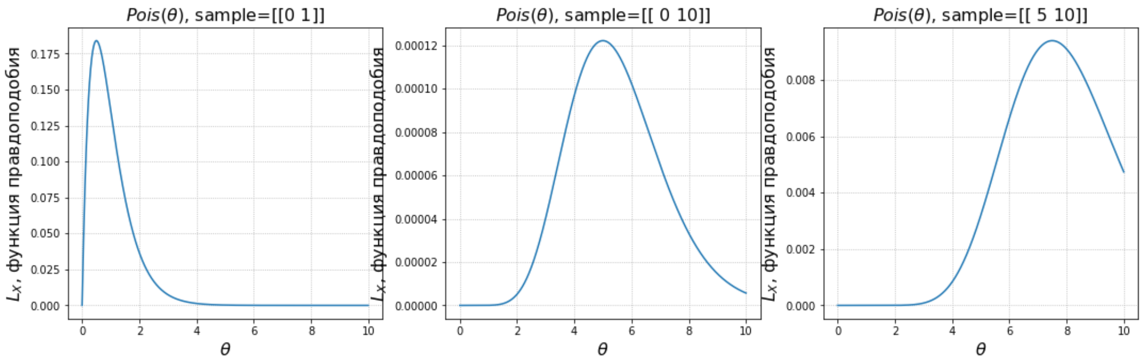
```
grid = np.linspace(0, 1, 1000).reshape((-1, 1))
draw_likelihood(sps.binom(n=5, p=grid).pmf, grid,
                [[0, 1], [5, 5], [0, 5]], '$Bin(5, \\theta)$')
```



e). Параметрическая модель $Pois(\theta)$, выборки: `[[0, 1], [0, 10], [5, 10]]`

In [48]:

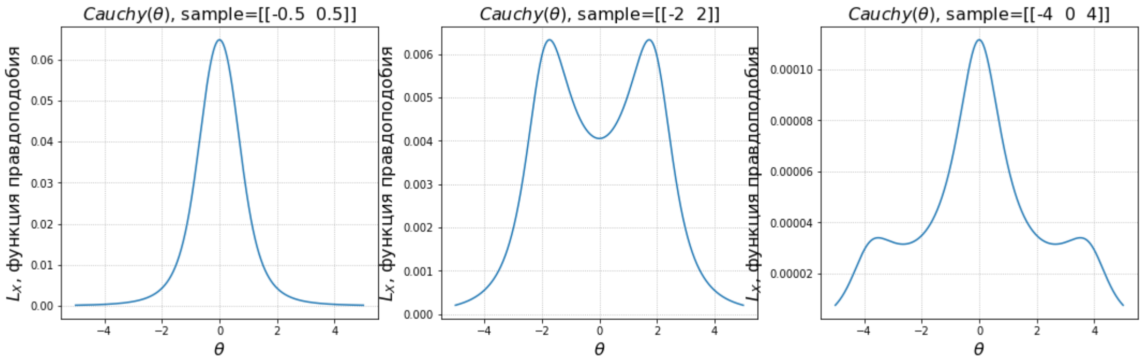
```
grid = np.linspace(0, 10, 1000).reshape((-1, 1))
draw_likelihood(sps.poisson(mu=grid).pmf, grid,
                [[0, 1], [0, 10], [5, 10]], '$Pois(\\theta)$')
```



f). Параметрическая модель $Cauchy(\theta)$, где θ — параметр сдвига, выборки: `[[-0.5, 0.5], [-2, 2], [-4, 0, 4]]`

In [56]:

```
grid = np.linspace(-5, 5, 1000).reshape((-1, 1))
draw_likelihood(sps.cauchy(loc=grid).pdf, grid,
                [[-0.5, 0.5], [-2, 2], [-4, 0, 4]], '$Cauchy(\\theta)$')
```



Сделайте вывод о том, как функция правдоподобия для каждой модели зависит от выборки. Является ли функция правдоподобия плотностью?

Вывод: Значения функции правдоподобия тем меньше, чем больше разброс значений выборки. Это верно для всех пунктов.

a). Для параметрической модели $\mathcal{N}(\theta, 1)$ максимум достигается при $\theta = \overline{X}$. При $|\theta - \overline{X}| > 2$ значения функции правдоподобия близки к нулю.

b) На занятиях было рассчитано, что максимум функции правдоподобия распределения $Exp(\theta)$ достигается при $\theta = 1/\overline{X}$. Максимум на графиках соответствует теоретическому максимуму.

c) Для параметрической модели $U[0, \theta]$ функция правдоподобия равна нулю при $\theta < X_{(n)}$. При $\theta = X_{(n)}$ она достигает своего максимума, потом убывает.

d) Для параметрической модели $Bin(5, \theta)$ функция правдоподобия достигает своего максимума при $\theta = \overline{X}$. Можно заметить, что она равна нулю при $\theta = 0, 1$, если выборка не состоит целиком из нулей или пятерок.

e) Для параметрической модели $Pois(\theta)$ функция правдоподобия достигает своего максимума при $\theta = \overline{X}$. Можно заметить, что она равна нулю при $\theta = 0$, если выборка не состоит целиком из нулей.

f) Для параметрической модели $Cauchy(\theta)$ функция правдоподобия достигает своего максимума при $\theta = \overline{X}$, если разброс значений в выборке не очень велик. Если выборка состоит из двух значений, то при $|X_1 - X_2| > 2$ у графика две точки максимума $\theta = X_1, X_2$. Если выборка состоит из трех значений. То максимум достигается рядом с $X_{(2)}$.

Функция правдоподобия не является плотностью, потому что это функция зависящая от параметра θ, в то время как плотность зависит от значения случайной переменной.

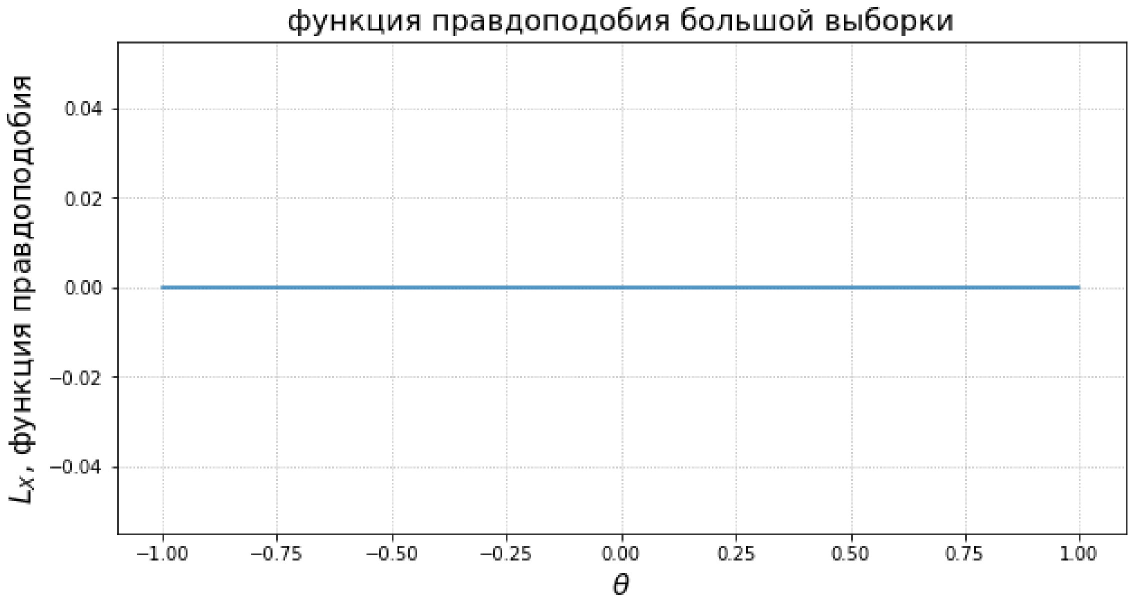
Сгенерируем выборку большого размера из стандартного нормального распределения и посчитаем ее функцию правдоподобия в модели $\mathcal{N}(\theta, 1)$. Выполните код ниже:

In [9]:

```
sample = sps.norm.rvs(size=10**3)
grid = np.linspace(-1, 1, 1000).reshape((-1, 1))
sample = np.array(sample)[np.newaxis, :]
```

In [10]:

```
plt.figure(figsize=(10, 5))
likelihood = np.prod(sps.cauchy(loc=grid).pdf(sample), axis=1)
plt.plot(grid, likelihood)
plt.xlabel('$\\theta$', fontsize=16)
plt.ylabel('$L_X$, функция правдоподобия', fontsize=16)
plt.grid(ls=':')
plt.title('функция правдоподобия большой выборки', fontsize=16)
plt.show()
```



Почему результат отличается от ожидаемого? Как обойти эту неприятность для подсчета оценки максимального правдоподобия? Реализуйте это.

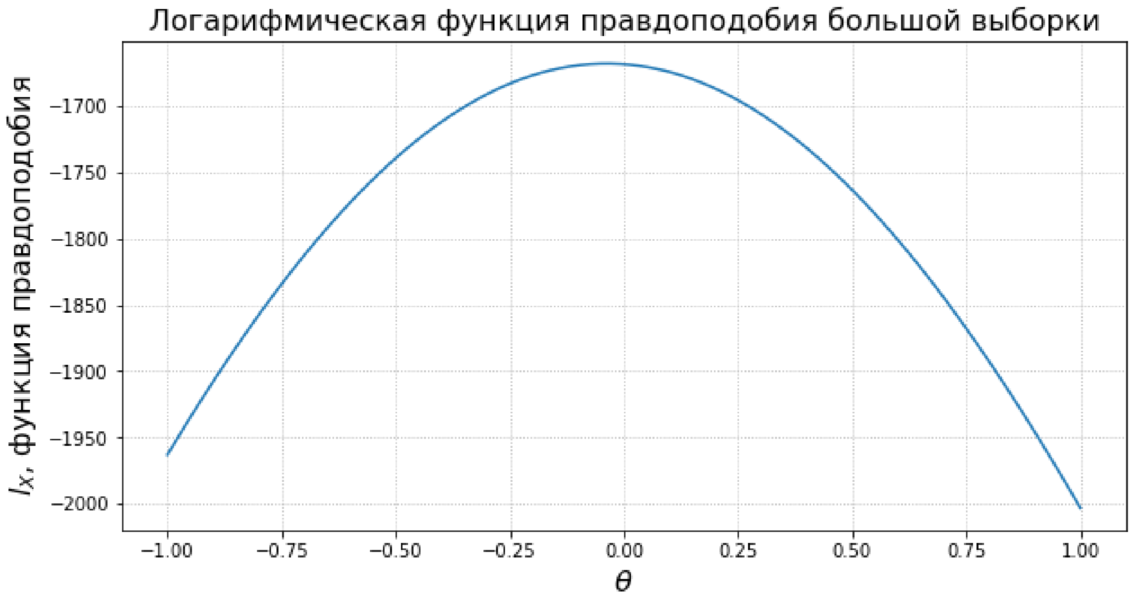
Подсказка: нужно использовать некоторый метод класса, реализующий это распределение

Ответ на вопрос и описание метода решения проблемы:

Проблема заключается в том, что для больших выборок функция правдоподобия оказывается слишком мала, равной нулю. Нужно использовать логарифмическую функцию правдоподобия.

In [11]:

```
plt.figure(figsize=(10, 5))
likelihood = np.sum(sps.cauchy(loc=grid).logpdf(sample), axis=1)
plt.plot(grid, likelihood)
plt.xlabel('$\\theta$', fontsize=16)
plt.ylabel('$l_X$, функция правдоподобия', fontsize=16)
plt.grid(ls=':')
plt.title('Логарифмическая функция правдоподобия большой выборки', fontsize=16)
plt.show()
```



На этом графике видно, что максимум функции правдоподобия достигается при $\theta = 0$. Следовательно, при больших выборках лучше находить максимум логарифмической функции правдоподобия.

Задача 6.

а). Пусть $X_1, \dots, X_n$ --- выборка из распределения $U[0, \theta]$. Рассмотрим оценки $2\overline{X}, (n+1)X_{(1)}, X_{(1)} + X_{(n)}, \frac{n+1}{n}X_{(n)}$. Вам необходимо сравнить эти оценки в равномерном подходе с квадратичной и линейной функциями потерь, построив графики функций риска при помощи моделирования.

Для каждого $\theta \in (0, 2]$ с шагом 0.01 сгенерируйте 5000 независимых выборок $X^1 = \left(X_1^1, \dots, X_{100}^1\right), \dots, X^{5000} = \left(X_1^{5000}, \dots, X_{100}^{5000}\right)$ из распределения $U[0, \theta]$.

Рассмотрим одну из перечисленных выше оценок $\hat{\theta}$. Посчитайте ее значение по каждой выборке. Тем самым, для данного θ получится 5000 реализаций этой оценки $\hat{\theta}_1, \dots, \hat{\theta}_{5000}$, где значение $\hat{\theta}_j$ посчитано по реализации выборки X^j.

Теперь можно оценить функцию риска этой оценки с помощью усреднения

$$\hat{R}_{\hat{\theta}}(\theta) = \frac{1}{5000}\sum_{j=1}^{5000} L\left(\hat{\theta}_j, \theta\right),$$

где L — одна из двух функций потерь: квадратичная $L(x, y) = (x-y)^2$ и линейная $L(x, y) = |x - y|$.

Для каждого из типов функций потерь постройте свой график. Нанесите на этот график для каждой из четырех оценок $\hat{\theta}$ оценку функции потерь $\hat{R}_{\hat{\theta}}(\theta)$, пользуясь шаблоном ниже. Ограничение сверху по оси *y* ставьте таким, чтобы графики функции риска с малыми значениями четко различались.

Совет: при тестировании кода запускайте его с небольшими размерами данных. Например, используйте 100 реализаций выборок. Финальные результаты получите, поставив требуемые значения размеров данных.

Решение:

In [40]:

```
grid=np.arange(0, 2, 0.01)
number_of_estimators = 5000
sample_size = 100
number_of_thetas = grid.size
sample = sps.uniform(scale=grid).rvs(size=(sample_size, number_of_estimators, number_of
_thetas))
```

In [41]:

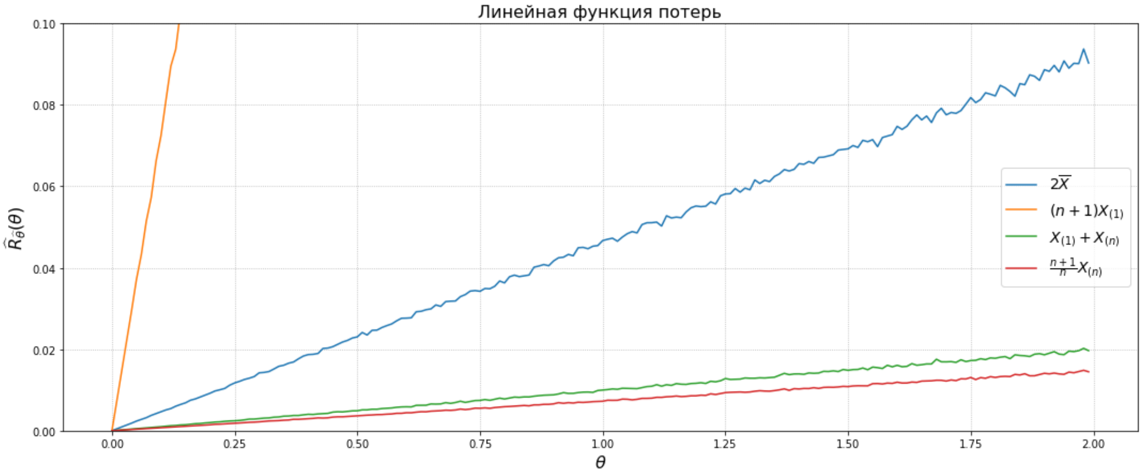
```
def estimate1(sample):
    "$2\\overline{X}$"
    return 2*np.mean(sample, axis=0)

def estimate2(sample):
    "$(n+1)X_{(1)}$"
    return np.min(sample, axis=0)*(sample_size+1)

def estimate3(sample):
    "$X_{(1)}+X_{(n)}$"
    return np.min(sample, axis=0) + np.max(sample, axis=0)

def estimate4(sample):
    "$\\frac{n+1}{n} X_{(n)}$"
    return np.max(sample, axis=0)*(sample_size+1)/sample_size
```

In [42]:

```
def loss_function1(x, y):
    "Линейная функция потерь"
    return abs(x-y)

def loss_function2(x, y):
    "Квадратичная функция потерь"
    return (x-y)**2

def risk_function(loss_function, estimators, theta_grid):
    "Функция риска"
    theta_table = np.full(estimators.shape, theta_grid)
    return np.mean(loss_function(theta_table, estimators), axis=0)

def draw_risk_functions(loss_function, estimator_list, ymin=0, ymax=1):
    plt.figure(figsize=(18,7))
    for estimator_function in estimator_list:
        estimator = estimator_function(sample)
        risk_function_values = risk_function(loss_function, estimator, grid)

        plt.plot(grid, risk_function_values,
             label=estimator_function.__doc__)
        plt.grid(ls=':')
        plt.xlabel('$\\theta$', fontsize=16)
        plt.ylabel('$\\widehat{R}_{\\widehat{\\theta}}(\\theta)$', fontsize=16)
        plt.legend(fontsize=14)
        plt.title(loss_function.__doc__, fontsize=16)
        plt.ylim((ymin, ymax))
    plt.show()
```

In [43]:

```
draw_risk_functions(loss_function1, [estimate1, estimate2, estimate3, estimate4], ymax=
0.1)
draw_risk_functions(loss_function2, [estimate1, estimate2, estimate3, estimate4], ymax=
0.015)
```



Сделайте вывод о том, какая оценка лучше и в каком подходе.

Вывод: В обоих подходах лучшая оценка $\frac{n+1}{n}X_{(n)}$

b). Пусть $X_1, \ldots, X_n$ --- выборка из распределения $Exp(\theta)$. Рассмотрим оценки $\left(k! \Big/ \overline{X^k}\right)^{1/k}$ для $1 \leqslant k \leqslant 5$, которые вы получили в домашнем задании. Проведите исследование, аналогичное пункту *a)*. Используйте цикл по k, чтобы не дублировать код. Функция факториала реализована как `scipy.special.factorial`.

Решение:

Рассмотрим значение функции риска на отрезке $[0.8, 1.2]$

In [92]:

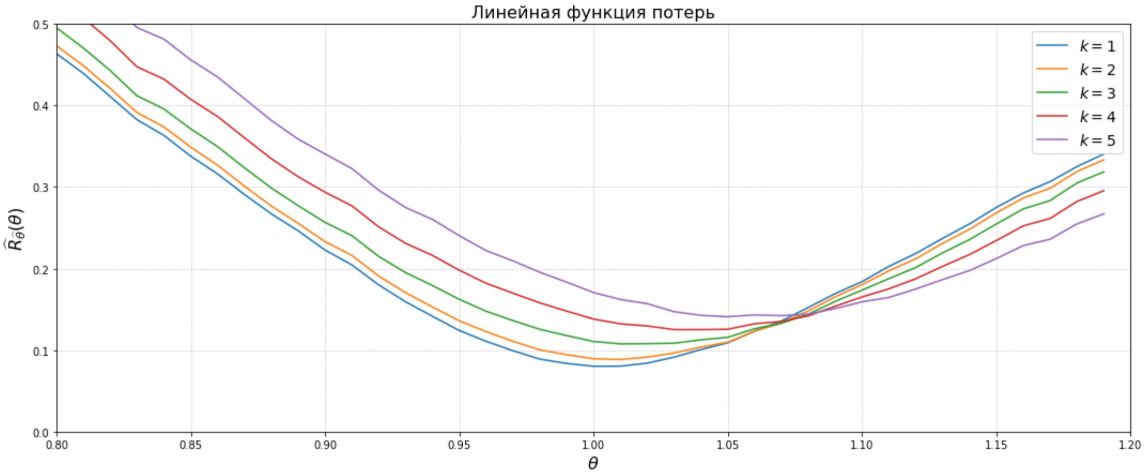
```
xmin, xmax = 0.8, 1.2
def estimate_expon(sample, k=1):
    kth_moment = np.mean(sample**k, axis=0)
    return (factorial(k)/kth_moment)**(1/k)

grid=np.arange(xmin, xmax, 0.01)
number_of_estimators = 5000
sample_size = 100
number_of_thetas = grid.size
sample = sps.expon(scale=grid).rvs(size=(sample_size, number_of_estimators, number_of_t
hetas))
```

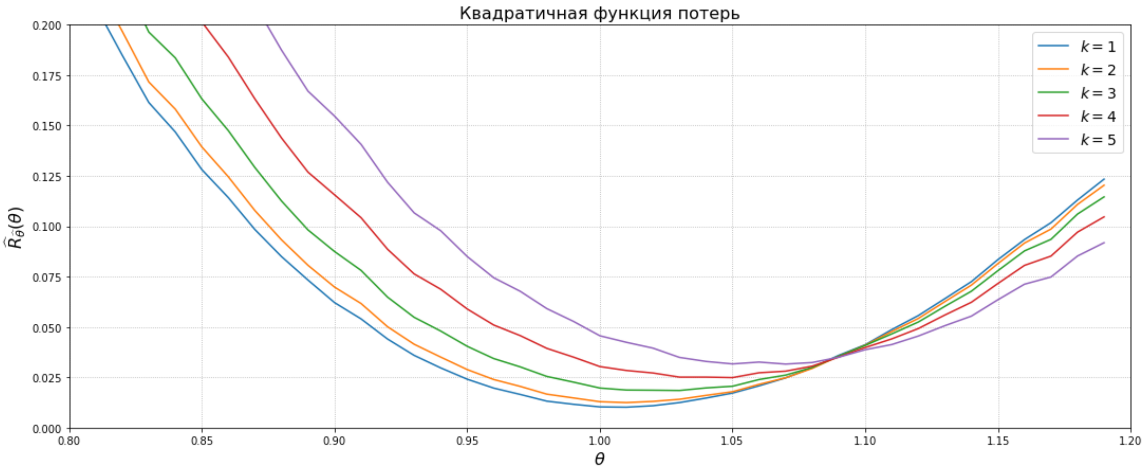
In [94]:

```
def draw_risk_functions(loss_function, estimator_function, ymin=0, ymax=1):
    plt.figure(figsize=(18,7))
    for k in range(1, 6):
        estimator = estimator_function(sample, k=k)
        risk_function_values = risk_function(loss_function, estimator, grid)

        plt.plot(grid, risk_function_values,
             label="$k=%d$"%k)
        plt.grid(ls=':')
        plt.xlabel('$\\theta$', fontsize=16)
        plt.ylabel('$\\widehat{R}_{\\widehat{\\theta}}(\\theta)$', fontsize=16)
        plt.legend(fontsize=14)
        plt.title(loss_function.__doc__, fontsize=16)
        plt.ylim((ymin, ymax))
        plt.xlim(xmin, xmax)
    plt.show()
```

In [95]:

```
draw_risk_functions(loss_function1, estimate_expon, ymax=0.5)
draw_risk_functions(loss_function2, estimate_expon, ymax=0.2)
```





Вывод: В обоих подходах нет наилучшей оценки, так как для обох подходов верно, что при $\theta \in [0.90, 1]$ наилучшая оценка достигается при $k = 1$, а при $\theta \in [1.10, 1.20]$ лучшая оценка достигается при при $k = 5$.