Статистика, прикладной поток

Практическое задание 5

В данном задании вы исследуете некоторые свойства доверительных интервалов и байесовских оценок.

Правила:

- Дедлайн **25 ноября 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 5`". Квадратные скобки обязательны. Если письмо дошло, придет ответ от автоответчика.
- Прислать нужно ноутбук и его pdf-версию (без архивов). Названия файлов должны быть такими: `5.N.ipynb` и `5.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 - 5 баллов **O2**
- Задача 2 - 5 баллов **O2**
- Задача 3 - 5 баллов **O2**
- Задача 4 - 5 баллов **O2**
- Задача 5 - 7 баллов **O2**
- Задача 6 - 7 баллов **O2**
- Задача 7 - 7 баллов **O2**
- Задача 8 - 10 баллов **O3**
- Задача 9 - 6 баллов **O2**

In [1]:
```
import numpy as np
import pandas as pd
import scipy.stats as sps
import matplotlib.pyplot as plt
import seaborn as sns
import warnings

sns.set()
warnings.filterwarnings("ignore")

%matplotlib inline
```

Доверительные интервалы

Задача 1.

В этой задаче нужно визуализировать доверительные интервалы для выборок из различных распределений.

Чтобы не плодить код, напишите следующую функцию (см. ниже). Пример построения есть в ноутбуке по `matplotlib`.

In [367]:
```
def draw_confidence_interval(
        left,  # левая граница интервала
        right,  # правая граница интервала
        estimation=None,  # если задана, то рисуется график оценки
        sample=None,  # если задано, то рисуются точки выборки
        ylim=(None, None),  # ограничение по оси y
        label='estimator'   # подпись
    ):
    grid = range(left.size)

    if sample is not None:
        plt.scatter(grid, sample, alpha=0.4, s=40, label='sample')

    if estimation is not None:
        plt.plot(grid, estimation, label=label)

    plt.fill_between(grid, left, right, alpha=0.5)

    plt.xlabel('sample size')
    plt.ylabel('$\\theta$')
    plt.ylim(ylim)
    plt.grid(True)
```

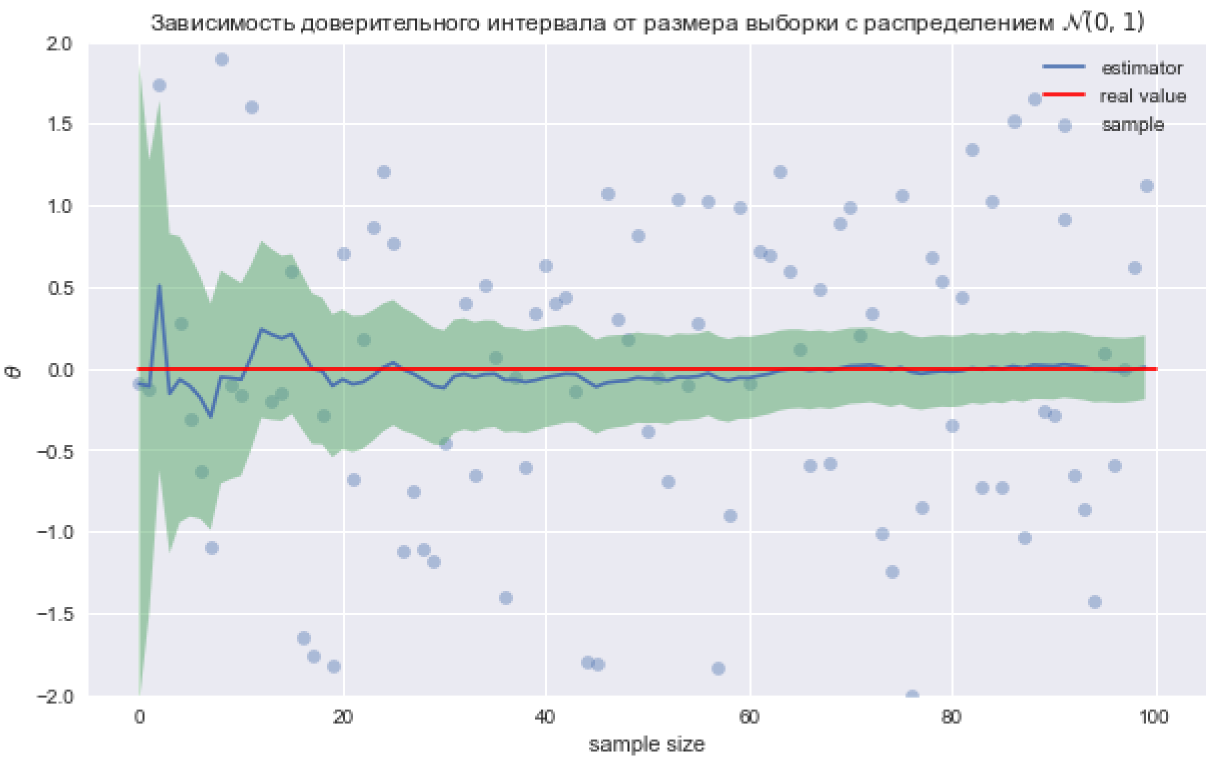
Рассмотрим следующие ситуации:

1. Выборка из распределения $\mathcal{N}(0, 1)$; точный доверительный интервал минимальной длины в параметрической модели $\mathcal{N}(\theta, 1)$.
2. Выборка из распределения $U[0, 1]$; точный доверительный интервал минимальной длины в параметрической модели $U[0, \theta]$ на основе статистики $X_{(n)}$.

Для каждой ситуации из перечисленных выше сгенерируйте выборку $X_1, \dots X_{100}$ и постройте график доверительных интервалов уровня доверия 0.95, вычисленных для всех подвыборок вида $X_1, \dots X_i$, $1 \le i \le 100$.

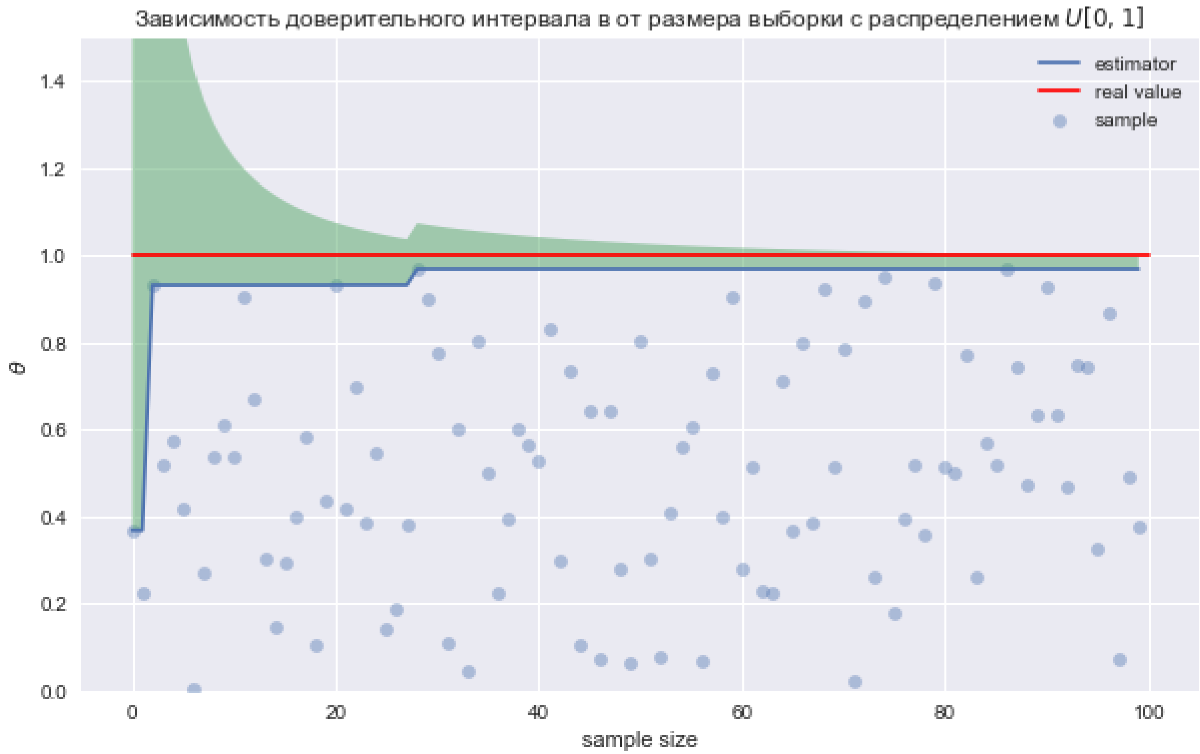
Постройте графики зависимости верхних и нижних границ интервала от размера выборки, используя написанную функцию. Нужно нанести на график точки выборки. Для вычисления квантилей у каждого распределения из `scipy.stats` используйте функцию `ppf`.

In [373]:
```
sample_size = 100
confidence_level = 0.95
grid = np.arange(1, sample_size+1)

sample = sps.norm.rvs(size=sample_size)
mean = sample.cumsum()/grid
z = sps.norm.ppf((1+confidence_level)/2)

left = mean - z/np.sqrt(grid)
right = mean + z/np.sqrt(grid)

plt.figure(figsize=(10, 6))
draw_confidence_interval(left, right, mean, sample=sample, ylim=(-2, 2))
plt.plot([0, sample_size], [0, 0], c='r', label='real value')
plt.title('Зависимость доверительного интервала от размера выборки с распределением $\\mathcal{N}(0, 1)$')
plt.legend();
```



In [376]:

```
sample = sps.uniform.rvs(size=sample_size)
maximum = np.maximum.accumulate(sample)
left = maximum
right = maximum/np.power(1-confidence_level, 1/grid)

plt.figure(figsize=(10, 6))
draw_confidence_interval(left, right, maximum, sample, (0, 1.5))
plt.plot([0, 100], [1, 1], c='r', label='real value')
plt.title('Зависимость доверительного интервала в от размера выборки с распределением $U[0, 1]$')
plt.legend();
```



Вывод: Реальное значение почти всегда попадает в построенные доверительные интервалы. Доверительный интервал для равномерного сходится намного быстрее доверительного интервала для нормального.

Задача 2.

Аналогично задаче 1 сгенерируйте выборку $X_1, \dots X_{100}$ из распределения $\Gamma(3, 2)$ и постройте доверительные интервалы для следующих случаев:

- точный асимптотический доверительный интервал в параметрической модели $\Gamma(\theta, 2)$; точки выборки наносить на график не нужно;
- точный асимптотический доверительный интервал для θ в параметрической модели $\Gamma(\theta, \beta)$, причем β неизвестно.

Изобразите интервалы *на одном* графике полупрозрачными цветами. Точки выборки наносить на график не нужно.

In [377]:
```
beta = 2
sample = sps.gamma(a=beta, scale=1/3).rvs(size=sample_size)

mean = sample.cumsum()/grid
tmp = np.sqrt(beta/grid)*sps.norm.ppf((1-confidence_level)/2)
left = 1/mean*(beta-tmp)
right = 1/mean*(beta+tmp)

plt.figure(figsize=(13, 8))
draw_confidence_interval(left, right, beta/mean, ylim=(1,5), label='estimator when beta is known ')
plt.plot([0, 100], [3, 3], c='r', label='real value')
plt.legend();
```

Сравните полученные интервалы.

Вывод: <...>

Задача 3.

Аналогично задаче 1 сгенерируйте выборку $X_1, \ldots X_{100}$ из стандартного распределения Коши и постройте доверительные интервалы для следующих случаев

- точный доверительный интервал минимальной длины в параметрической модели $\mathcal{N}(\theta, 1)$;
- точный асимптотический доверительный интервал в параметрической модели распределения Коши со сдвигом, используя выборочную медиану;
- точный асимптотический доверительный интервал в параметрической модели распределения Коши со сдвигом, используя асимптотически эффективную оценку.

Изобразите интервалы *на одном* графике полупрозрачными цветами. Точки выборки нужно нанести на график.

С занятий известно, что выборочная медиана это асимтпотически нормальная оценка сдвига распределения Коши с асимпотической дисперсией $\sigma^2 = \frac{\pi^2}{4}$.

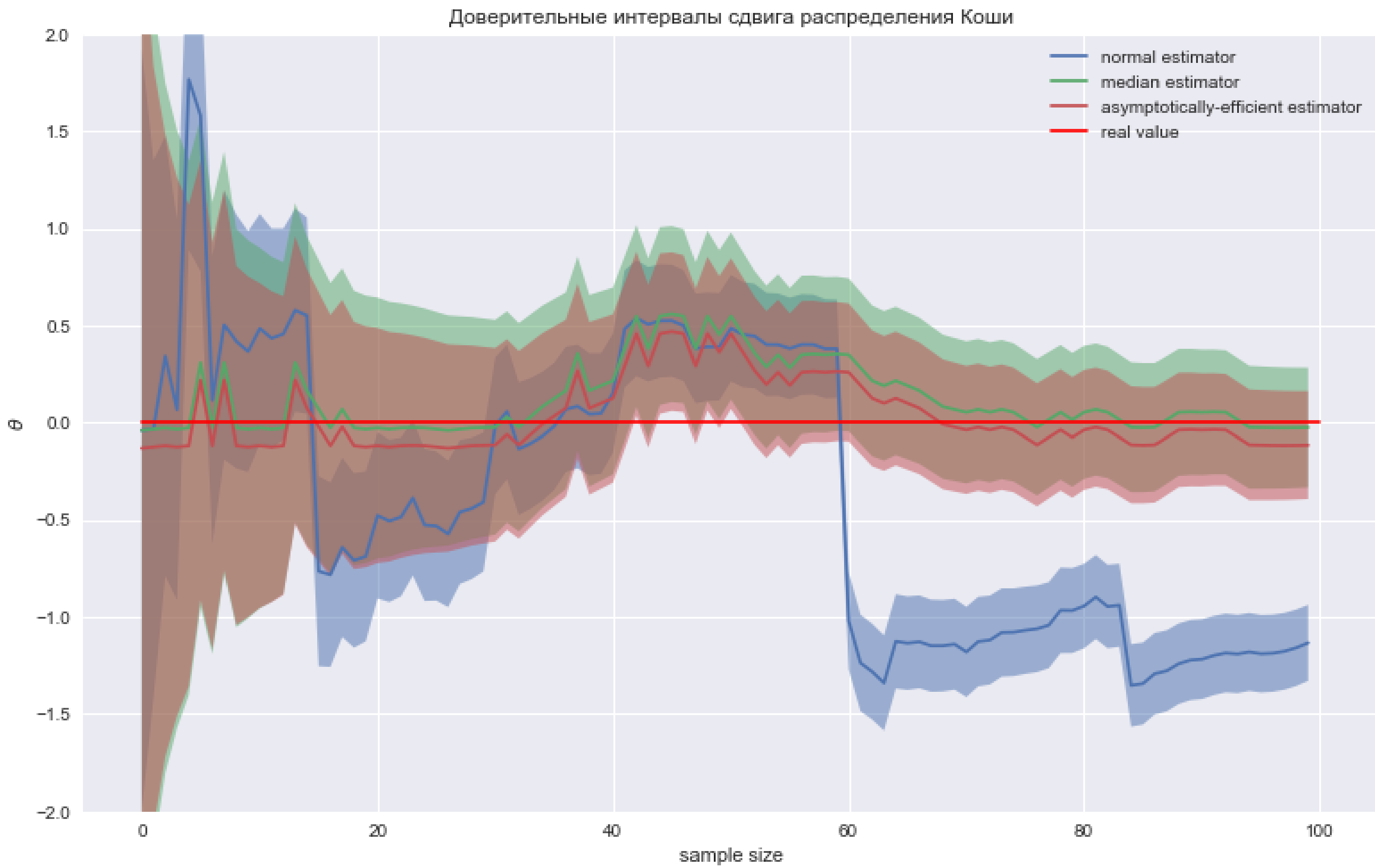
Асимптотически эффективная оценка

$$\hat{\theta} = \hat{\mu} + \frac{\sum \frac{X_i - \mu}{1 + (X_i - \mu)^2}}{\sum \frac{1 - (X_i - \mu)^2}{\left(1 + (X_i - \mu)^2\right)^2}}$$

имеет асимптотическую дисперсию $\sigma^2 = 2$.

In [382]:

```
sample = sps.cauchy.rvs(size=sample_size)

mean = sample.cumsum()/grid
z = sps.norm.ppf((1+confidence_level)/2)

left = mean - z/np.sqrt(grid)
right = mean + z/np.sqrt(grid)

plt.figure(figsize=(13, 8))
draw_confidence_interval(left, right, mean, ylim=(-2, 2), label='normal estimator')

median = [np.median(sample[0:i]) for i in grid]
u = sps.norm(scale=3.14/2).ppf((1-confidence_level)/2)
draw_confidence_interval(median-u/np.sqrt(grid), median+u/np.sqrt(grid), median, label='median estimator')

x = sample-median
numerator = x/(1+x**2)
denominator = (1-x**2)/(1+x**2)**2

estimator = median + numerator.sum()/denominator.sum()

u = sps.norm(scale=2**(1/2)).ppf((1-confidence_level)/2)
draw_confidence_interval(estimator-u/np.sqrt(grid), estimator+u/np.sqrt(grid), estimator,
                         label='asymptotically-efficient estimator')

plt.plot([0, 100], [0, 0], c='r', label='real value')
plt.title('Доверительные интервалы сдвига распределения Коши')
plt.legend();
```



Сравните полученные интервалы.

Вывод: Доверительный интервал посчитанный для нормального распределения часто не содержит истиного значения. Потому что некорректно оценивать θ средним, в то время как у распределения Коши не существует матожидания.

Два других доверительных интервала гораздо лучше, они почти всегда содержат истиное значение. Но интервал для ассимтически эффективной оценки чуть уже.

Задача 4.

Пусть $X_1, \ldots, X_n$ --- выборка из распределения $\mathcal{N}(a, \sigma^2)$. Постройте точную доверительную область для параметра $\theta = (a, \sigma)$ уровня доверия $\alpha = 0.95$ для сгенерированной выборки размера $n \in \{5, 20, 50\}$ из стандартного нормального распределения.

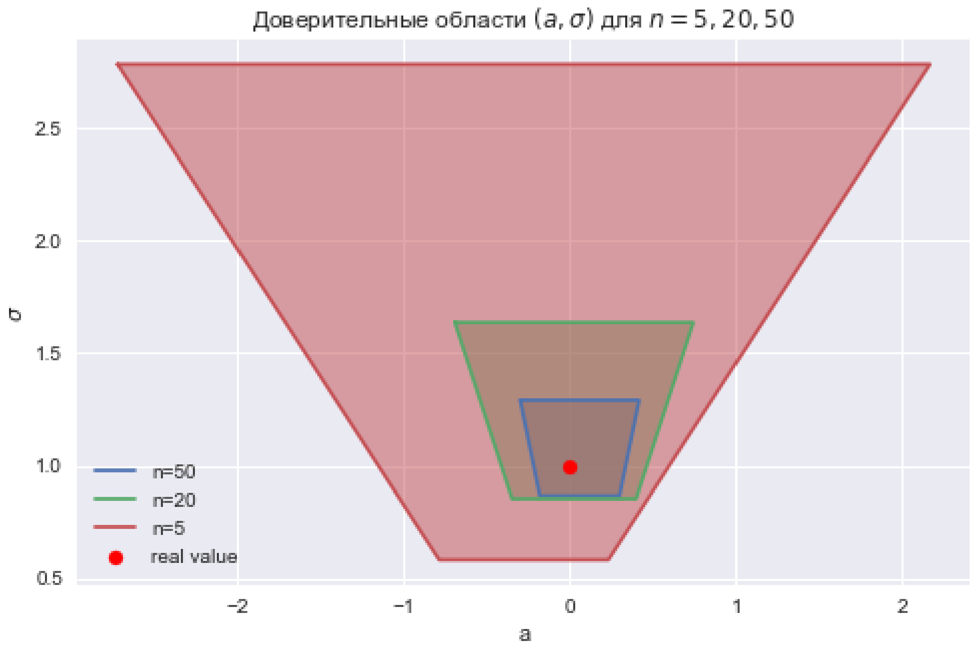
Из домашней задачи известно, что область задается системой уравнений:

$$\begin{cases} \sqrt{\frac{nS^2}{u_{\frac{1+\alpha}{2}}}} < \sigma < \sqrt{\frac{nS^2}{u_{\frac{1-\alpha}{2}}}} \\ |a - \overline{X}| < \frac{z_{\frac{1+\alpha}{2}}\sigma}{\sqrt{n}} \end{cases}$$

где u_α это α квантиль распределения $\chi^2(n-1)$

In [391]:
```
def draw_convidence_domain(sample, confidence_level=0.95):
    mean = sample.mean()
    z = sps.norm.ppf((1+confidence_level)/2)
    S2 = np.var(sample)
    u1 = sps.chi2(df=n-1).ppf((1+confidence_level)/2)
    min_sigma = np.sqrt(n*S2/u1)
    max_sigma = np.sqrt(n*S2/sps.chi2(df=n-1).ppf((1-confidence_level)/2))

    v_a = [mean-z*max_sigma/np.sqrt(n),      # абсциссы точек
           mean-z*min_sigma/np.sqrt(n),
           mean+z*min_sigma/np.sqrt(n),
           mean+z*max_sigma/np.sqrt(n)]

    v_sigma = [max_sigma, min_sigma, min_sigma, max_sigma]   # ординаты точек

    plt.plot(v_a+[v_a[0]], v_sigma+[v_sigma[0]], label='n=%d'%n)
    plt.fill_between(v_a, v_sigma, max_sigma*np.ones(4), alpha=0.5)

    plt.xlabel('a')
    plt.ylabel('$\sigma$')
```

In [392]:
```
plt.figure(figsize=(8, 5))
sample = sps.norm.rvs(size=50)

for n in [50, 20, 5]:
    draw_convidence_domain(sample[:n])

plt.title('Доверительные области $(a, \sigma)$ для $n=5, 20, 50$')
plt.scatter([0], [1], c='r', label='real value')
plt.legend();
```



Вывод: Реальное значение почти содержится в доверительных областях. Чем больше n тем меньше площадь доверительной области.

Задача 5.

В данном задании вам нужно изучить доверительные интервалы для параметра сдвига в нормальной модели в случае неизвестной дисперсии. Требуется построить асимтотический доверительный интервал (через центральную предельную теорему и лемму Слуцкого) и точный неасимптотический (через распределения хи-квадрат и Стьюдента).

Вывод этих интервалов был разобран на семинарах. Выпишите только ответы.

Асимптотический доверительный интервал: $\left(\bar{X} - z_{\frac{1+\alpha}{2}} \frac{S}{\sqrt{n}}, \bar{X} + z_{\frac{1+\alpha}{2}} \frac{S}{\sqrt{n}}\right)$

Точный доверительный интервал: $\left(\bar{X} - t_{\frac{1+\alpha}{2}, n-1} \frac{S}{\sqrt{n}}, \bar{X} + t_{\frac{1+\alpha}{2}, n-1} \frac{S}{\sqrt{n}}\right)$

Реализуйте функции построения этих интервалов по выборке.

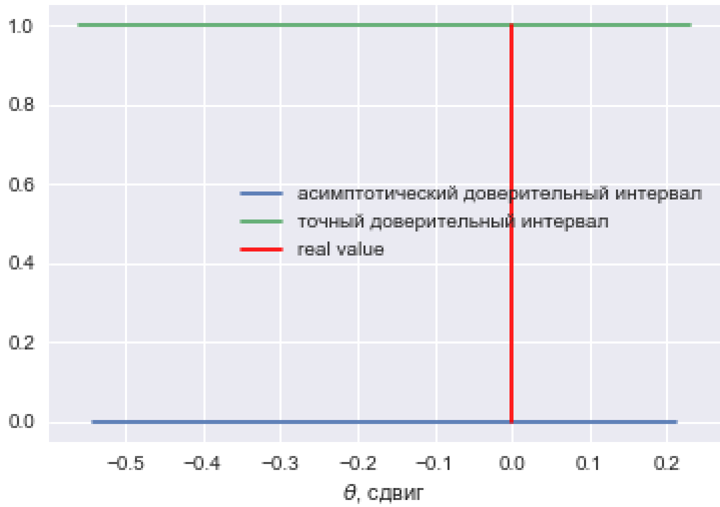
In [209]:

```
def calculate_asymptotic_confidence_interval(sample, alpha=0.95):
    z = sps.norm.ppf((1+alpha)/2)
    mean = sample.mean()
    S = np.sqrt(sample.var())
    d = z*S/np.sqrt(sample.size)
    return (mean-d, mean+d)

def calculate_confidence_interval(sample, alpha=0.95):
    n = sample.size
    t = sps.t(df=n-1).ppf((1+alpha)/2)
    mean = sample.mean()
    S = np.sqrt(sample.var())
    d = t*S/np.sqrt(n)
    return (mean-d, mean+d)
```

Сгенерируйте выборку из нормального расределения и сравните два доверительных интервала в зависимости от размера выборки. Для сравнения отобразите оба интервала на одном графике. Поясните теоретическую причину такого поведения доверительных интервалов.

Указание: рассматривайте длину выборки около 20-30.

In [388]:
```
sample = sps.norm.rvs(size=30)
asymptotic_confidence_interval = calculate_asymptotic_confidence_interval(sample)
accurate_confidence_interval = calculate_confidence_interval(sample)
plt.plot(asymptotic_confidence_interval, [0, 0], label='асимптотический доверительный интервал')
plt.plot(accurate_confidence_interval, [1, 1], label='точный доверительный интервал')
plt.plot([0,0], [0,1], c='r', label='real value')
plt.xlabel('$\\theta$, сдвиг')
plt.legend();
```



Вывод: Асимптотиеский доверительный интервал чуть уже. Видимо потому что его вероятность стремится к альфе снизу.

Скачайте данные [`wine dataset` (http://archive.ics.uci.edu/ml/datasets/wine)](http://archive.ics.uci.edu/ml/datasets/wine) и рассмотрите столбцы `Alcalinity of ash`, `Nonflavanoid phenols`, `Proanthocyanins` и `Hue` для вина первого типа (за тип вина отвечает первый столбец).

Постройте доверительные интервалы для параметров сдвига каждого из столбцов, предполагая, что столбцы имеют нормальное распределение. Нужно построить доверительные интервалы обоих рассмотренных выше типов. Запишите их в виде таблицы.

In [394]:
```
names= ['Cultivar', 'Alcohol', 'Malic acid', 'Ash', 'Alcalinity of ash', 'Magnesium', 'Total phenols', 'Flavanoids',
        'Nonflavanoid phenols', 'Proanthocyanins', 'Color intensity', 'Hue', 'OD280/OD315 of diluted wines', 'Proline'
]
sample = pd.read_csv('wine.data', names=names)

names = ["Alcalinity of ash", "Nonflavanoid phenols", "Proanthocyanins", "Hue"]
sample = sample[sample.Cultivar==1][names]
```

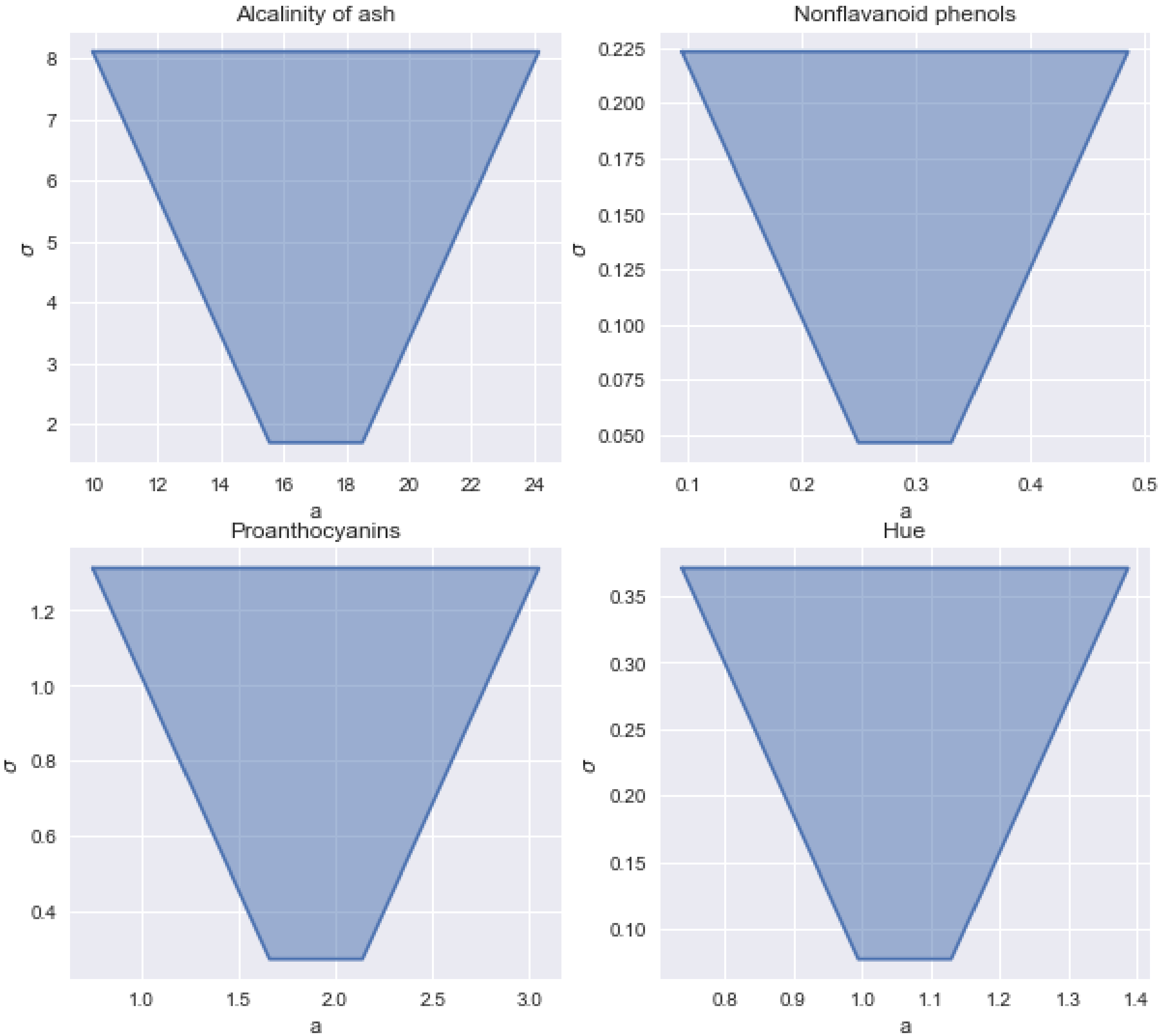
In [395]:
```
confidence_intervals = pd.DataFrame([[calculate_asymptotic_confidence_interval(sample[name]),
                                     calculate_confidence_interval(sample[name])] for name in names],
                                    index=names, columns=['asymptotic_confidence_interval', 'accurate_confidence_inter
val'])
confidence_intervals
```

Out[395]:

	asymptotic_confidence_interval	accurate_confidence_interval
Alcalinity of ash	(16.387554298307265, 17.68702197287918)	(16.37371289050777, 17.700863380678676)
Nonflavanoid phenols	(0.2721258443172716, 0.30787415568272836)	(0.27174506766123374, 0.3082549323387662)
Proanthocyanins	(1.7941659452565304, 2.0044781225400805)	(1.791925784406095, 2.0067182833905157)
Hue	(1.0323115379574475, 1.091756258652722)	(1.0316783566657761, 1.0923894399443934)

Наконец, постройте точную доверительную область для параметров сдвига и масштаба для каждого из рассматриваемых столбцов. Для экономии места стройте графики в два столбца.

In [399]:
```
plt.figure(figsize=(10, 9))
for i in range(4):
    plt.subplot(2, 2, i+1)
    draw_convidence_domain(sample[names[i]])
    plt.title(names[i])
```



Вывод: Для этих выборок точный и асимптотический интервалы также практический не различаются.

Доверительные соответствуют интервалам.

Байесовский подход

Задача 6.

Пусть $X_1, \ldots, X_n$ --- выборка из распределения $\mathcal{N}(\theta, \sigma^2)$, а параметр θ в качестве априорного распределения имеет стандартное распределение Коши со сдвигом.

Апостериорное распределение вычисляется по формуле:

$$q\,(t \mid\, x) = \frac{q(t)p_t(x_1)\ldots p_t(x_n)}{\int\limits_{\Theta} q(t)p_t(x_1)\ldots p_t(x_n)dt},$$

где $p_t(x)$ --- плотность распределения $\mathcal{N}(t, \sigma^2)$, а $q(t)$ --- плотность распределения Коши. Как было сказано на лекции, аналитически интеграл в знаменателе посчитать не удастся. Однако, этот интеграл можно вычислить численно, например, с помощью метода Монте-Карло.

В данном случае, интеграл $\int\limits_{\mathbb{R}} f(x)p(x)dx$, где $p(x)$ -- некоторая плотность, можно оценить как $\frac{1}{k}\sum\limits_{j=1}^{k} f(\xi_i)$, где $\xi_1, \ldots, \xi_k$ -- сгенерированная выборка из распределения, имеющего плотность $p(x)$.

Рассмотрим столбец `Alcalinity of ash` датасета [о вине (http://archive.ics.uci.edu/ml/datasets/Wine)](http://archive.ics.uci.edu/ml/datasets/Wine).

Для выборки, образованной эти столбцом посчитайте c -- знаменатель в формуле Байеса. Параметры априорного распределения выберите некоторым разумном способом, не опираясь на данные. Какой размер вспомогательной выборки в методе приближенного интегрирования необходим, чтобы с большой точностью посчитать значение c?

In [339]:

```
alcanity = sample[[names[0]]]
#apr = sps.norm(loc=17, scale=6**(1/2))
apr = sps.cauchy(loc=17)
ksi = apr.rvs(size=100000)
c = sps.norm(loc=ksi).pdf(alcanity).prod(axis=0).mean()
print('c =', c)
```

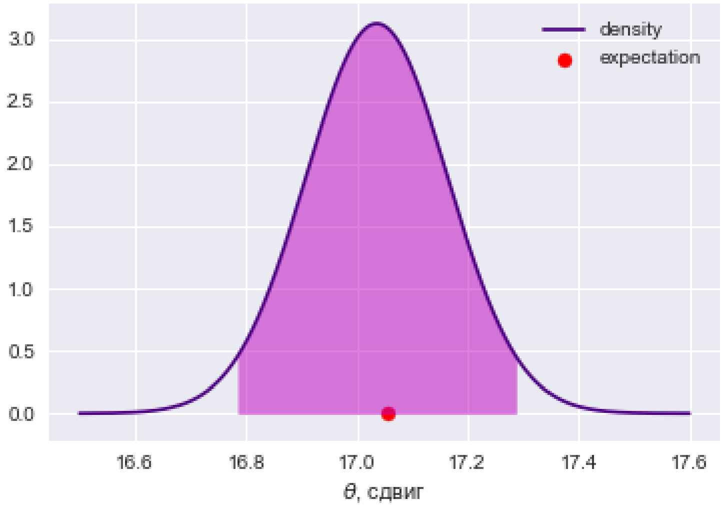
```
c = 6.334117834705507e-107
```

Для апостериорного распределения:

- Нарисуйте график плотности;
- Посчитайте математическое ожидание;
- Найдите симметричный 95%-ый доверительный интервал.

In [400]:

```python
grid = np.linspace(16.5, 17.6, 1000)

def find_density(grid):
    return apr.pdf(grid)*sps.norm(loc=grid).pdf(alcanity).prod(axis=0)/c
density = find_density(grid)
expectation = 1.1*np.mean(grid*density)

plt.plot(grid, density, label='density', color='indigo')
plt.scatter([expectation], [0], label='expectation', color='r')

def quantile(grid, density, alpha):
    # находит альфа-квантиль распределения с плотностью density
    norm = density.sum()
    distr = sps.rv_discrete(values=(grid,density/norm))
    return distr.ppf(alpha)

conf_int = quantile(grid, density, [0.05/2, 1-0.05/2])
new_grid = np.linspace(conf_int[0], conf_int[1], 1000)
plt.fill_between(new_grid, np.zeros(1000), find_density(new_grid), alpha=0.5, color='m')
plt.xlabel('$\\theta$, сдвиг')
plt.legend();
```



Вывод: хороший доверительный интервал и выборка интересная

Задача 7.

Рассмотрим схему испытаний Бернулли (т.е. броски монет) с вероятностью успеха p.

Постройте несколько графиков сопряженного распределения для разных параметров и охарактеризуйте, как его значения параметров соотносятся с априорными знаниями о монете. Это могут быть, например, знания вида

- монета скорее честная (при таком априорном распределении наиболее вероятны значения p в окрестности 0.5);
- монета скорее нечестная, перевес неизвестен (наименее вероятны значения p в окрестности 0.5);
- монета скорее нечестная, перевес в сторону герба (наиболее вероятны значения p в окрестности 1);
- монета скорее честная, либо с небольшим перекосом вправо (наиболее вероятны значения p в окрестности ~0.6);
- ничего не известно (все значения равновероятны).

Для каждого случая из перечисленных выше постройте график плотности сопряженного распределения (на одной фигуре).

In [ ]:
```
<...>
```

Вывод: <...>

Ниже приведена реализация некоторых вспомогательных функций.

In [ ]:

```
def draw_posteriori(grid, distr_class, post_params, xlim=None):
    ''' Рисует серию графиков апостериорных плотностей.
        grid --- сетка для построения графика
        distr_class --- класс распределений из scipy.stats
        post_params --- параметры апостериорных распределений
                        shape=(размер выборки, кол-во параметров)
    '''

    size = post_params.shape[0] - 1

    plt.figure(figsize=(12, 7))
    for n in range(size+1):
        plt.plot(grid,
                 distr_class(post_params[n]).pdf(grid) \
                     if np.isscalar(post_params[n]) \
                     else distr_class(*post_params[n]).pdf(grid),
                 label='n={}: {}'.format(n, post_params[n]),
                 lw=2.5,
                 color=(1-n/size, n/size, 0))
    plt.grid(ls=':')
    plt.legend()
    plt.xlim(xlim)
    plt.show()


def draw_estimations(ml, distr_class, post_params, confint=True, ylim=None):
    ''' Рисует графики байесовской оценки (м.о. и дов. инт.) и ОМП.
        ml --- Оценка максимального правдоподобия для 1 <= n <= len(sample)
        distr_class --- класс распределений из scipy.stats
        post_params --- параметры апостериорных распределений
                        shape=(размер выборки, кол-во параметров)
    '''

    size = len(ml)
    distrs = []
    for n in range(size+1):
        distrs.append(distr_class(post_params[n]) \
                          if np.isscalar(post_params[n]) \
                          else distr_class(*post_params[n]))

    plt.figure(figsize=(12, 4))
```

```
    plt.plot(np.arange(size+1), [d.mean() for d in distrs],
             label='Bayes', lw=1.5)
    plt.fill_between(np.arange(size+1), [d.ppf(0.975) for d in distrs],
                     [d.ppf(0.025) for d in distrs], alpha=0.1)
    plt.plot(np.arange(size)+1, ml, label='MLE', lw=1.5)
    plt.grid(ls=':')
    plt.ylim(ylim)
    plt.legend()
    plt.show()
```

Реализуйте следующую функцию

In [ ]:
```
def bern_posterior_params(sample, a, b):
    ''' Возвращает параметры апостериорного распределения
        для всех 0 <= n <= len(sample).
        a, b --- параметры априорного распределения.
    '''


    <...>
    return params
```

Проведите по 15 бросков симметричной и несимметричной монет (можно сгенерировать) и рассмотрите для каждой из них два случая --- параметры априорного распределения подобраны правильно или неправильно. Постройте графики, воспользовавшись функциями `draw_posteriori` и `draw_estimations`.

In [ ]:
```
<...>
```

Сделайте вывод. Что можно сказать про зависимость от параметров априорного распределения? Сравните байесовские оценки с оценкой максимального правдоподобия.

Вывод: <...>

Задача 8.

Один экзаменатор на экзамене по математической статистике при выставлении оценки студенту пользуется следующей схемой. В течении экзамена экзаменатор задает студенту некоторое количество вопросов, получая тем самым выборку $X_1, \ldots, X_n \sim Bern(p)$ --- индикаторы того, что студент на вопрос ответил правильно. При этом сначала он подбирает некоторое априорное распределение на основе его знаний о студенте к моменту начала ответа. После каждого ответа студента экзаменатор вычисляет апостериорное распределение и строит байесовский доверительный интервал для p уровня доверия 0.95. Если после очередного ответа студента доверительный интервал содержит лишь одно число $i/10$, где $i \in \{0, \ldots, 10\}$, то экзаменатор выставляет студенту оценку $i + 1$.

Ответьте на следующие вопросы:

- Квантили какого уровня нужно выбирать экзаменатору при построении доверительного интервала, чтобы задавать студенту как можно меньше вопросов? Какие оценки будет выставлять экзаменатор в таком случае?
- Как зависит оценка студента и среднее количество заданных вопросов у различных студентов (по уровню знаний) при различных априорных представлений экзаметора?

In [ ]:
```
<...>
```

Вывод: <...>

Задача 9.

Проведите исследование, аналогичное задаче 7 для выборок из распределений

- $\mathcal{N}(\theta, 1)$
- $Exp(\theta)$

In [ ]:
```
<...>
```

Вывод: <...>