СКБ201 Тур Тимофей

Теория вероятности, долгосрочные домашние задания. Вариант 68: дискретное - 3, непрерывное - 5.

3 - Дискретное равномерное 1: $P(x) = \theta^{-1}, x \in \{1, \dots, \theta\}, \theta = 29$

Обозначим дискретное распределение в дальнейшем за ξ

5 - Треугольное: $f(x) = \begin{cases} \dfrac{2x}{\theta}, & \text{если } x \in [0, \theta] \\ \dfrac{2(1-x)}{1-\theta}, & \text{если } x \in (\theta, 1] \\ 0, & \text{иначе} \end{cases}, \theta = 0.45$

Обозначим абсолютно непрерывное распределение в дальнейшем за η

Навигация

Домашнее задание 1

Дискретное

Задание 1

Функция распределения

$$F(n) \stackrel{\text{def}}{=} P(\xi \leq n) = \sum_{k=1}^{n} P(\xi = k) = \sum_{k=1}^{n} \theta^{-1} = \underline{n\theta^{-1}}$$

Математическое ожидание

$$M\xi \stackrel{\text{def}}{=} \sum_{x=1}^{\theta} xP(\xi = x) = \sum_{x=1}^{\theta} x\theta^{-1} = \theta^{-1} \sum_{x=1}^{\theta} x = \theta^{-1} \frac{1+\theta}{2}\theta = \underline{\frac{\theta+1}{2}}$$

Дисперсия

$$D\xi \stackrel{\text{def}}{=} M(\xi - M\xi)^2 = M\xi^2 - (M\xi)^2$$

$$M\xi^2 \stackrel{\text{def}}{=} \sum_{x=1}^{\theta} x^2 P(\xi = x) = \theta^{-1} \sum_{x=1}^{\theta} x^2 = \theta^{-1} \frac{\theta(1-\theta)(1+2\theta)}{6} = \frac{(1+\theta)(1+2\theta)}{6}$$

$$\Rightarrow D\xi = M\xi^2 - (M\xi)^2 = \frac{(1+\theta)(1+2\theta)}{6} - (\frac{\theta+1}{2})^2 = \frac{2(1+3\theta+2\theta^2) - 3(1+2\theta+\theta^2)}{12} = \underline{\frac{\theta^2-1}{12}}$$

Квантиль уровня γ

$$P(\xi \leq x_\gamma) \geq \gamma \Leftrightarrow \sum_{k=1}^{x_\gamma} P(\xi = k) \geq \gamma \Leftrightarrow \sum_{k=1}^{x_\gamma} \theta^{-1} \geq \gamma \Leftrightarrow x_\gamma \theta^{-1} \geq \gamma \Leftrightarrow x_\gamma \geq \gamma\theta \Rightarrow \underline{x_\gamma = \gamma\theta}$$

Задание 2

Примером события с дискретным равномерным распределением может быть игра "Bingo". Но не вся она, а лишь ее часть. В ней, подобно лото, участникам выдаются цветные листки с числами и маркерами, а ведущий стоит у аппарата, поторый по нажатию кнопки выдает случайный шарик, крутящийся в нем. Шарик имеет цвет и номер, и участники выделяют соответсвующие ячейки на своем листе, пока у них не получатся какая-нибудь соответствующая последовательность. (Лично я увидел эту игру в сериале "Лучше звоните Солу" в первом сезоне). Чтобы эта модель была применима к нашему распределению, игру следует упростить: На листке всего 1 номер и мячики не имеют цвета. Тогда шанс появления какого-то мячика будет равен $\frac{1}{\text{количество мячиков} = \theta} = \theta^{-1}$, и, соответственно шанс выигрыша какого-то игрока тоже равен θ^{-1}

Задание 3

Поделим отрезок $[0, 1]$ на сегменты равные θ^{-1}. Их будет в точности θ штук, а выборка определяется вхождением в какой из последовательных отрезков получилось у случайной величины: $\sqsupset u$ - сгенерированная равномерно распределенная величина на отрезке $[0, 1]$, тогда x определяется по формуле $(x-1)\theta^{-1} \leq u < x\theta^{-1}$

In [1]:
```python
import numpy as np

def generate_xi(theta=29):
    rng = np.random.default_rng()
    u = rng.uniform()
    for k in range(1, theta + 1):
        if (k - 1) / theta <= u < k / theta:
            return k
```

Абсолютно непрерывное

Задание 1

Функция распределения

$$F(x) \overset{\text{def}}{=} \int_{\mathbb{R}} f(x) = \begin{cases} 0, & \text{если } x < 0 \\ \int_0^x \frac{2t}{\theta} dt, & \text{если } x \in [0, \theta] \\ \int_0^\theta \frac{2t}{\theta} dt + \int_\theta^x \frac{2(1-t)}{1-\theta} dt, & \text{если } x \in (\theta, 1] \\ \int_0^\theta \frac{2t}{\theta} dt + \int_\theta^1 \frac{2(1-t)}{1-\theta}, & \text{если } x > 1 \end{cases}$$

$$1) \int_0^x \frac{2t}{\theta} dt = \frac{1}{\theta} \int_0^x 2t dt = \frac{1}{\theta} t^2 \Big|_0^x = \frac{1}{\theta} x^2$$

$$2) \int_0^\theta \frac{2t}{\theta} dt + \int_\theta^x \frac{2(1-t)}{1-\theta} dt = \theta + \frac{2}{1-\theta} \int_\theta^x (1-t) dt = \theta + \frac{2}{1-\theta} (t - \frac{1}{2} t^2) \Big|_\theta^x =$$

$$= \theta + \frac{2}{1-\theta} (x - \frac{1}{2} x^2 - \theta + \frac{1}{2} \theta^2) = \theta + \frac{1}{1-\theta} (2x - x^2 - 2\theta + \theta^2) =$$

$$= \frac{1}{1-\theta} (2x - x^2 - \theta)$$

$$3) \int_0^\theta \frac{2t}{\theta} dt + \int_\theta^1 \frac{2(1-t)}{1-\theta} = \frac{1}{1-\theta} (2 - 1 - \theta) = \frac{1-\theta}{1-\theta} = 1$$

$$\Rightarrow F(x) = \begin{cases} 0, & \text{если } x < 0 \\ \frac{1}{\theta} x^2, & \text{если } x \in [0, \theta] \\ \frac{1}{1-\theta} (2x - x^2 - \theta), & \text{если } x \in (\theta, 1] \\ 1, & \text{если } x > 1 \end{cases}$$

Математическое ожидание

$$M\eta \stackrel{\text{def}}{=} \int_{\mathbb{R}} f(x)xdx = \int_0^\theta \frac{2x}{\theta}xdx + \int_\theta^1 \frac{2(1-x)}{1-\theta}xdx = \frac{2}{\theta}\int_0^\theta x^2dx + \frac{2}{1-\theta}\int_\theta^1 (x-x^2)dx =$$
$$= \frac{2}{3\theta}x^3\Big|_0^\theta + \frac{2}{1-\theta}(\frac{1}{2}x^2 - \frac{1}{3}x^3)\Big|_\theta^1 = \frac{2}{3\theta}\theta^3 + \frac{2}{1-\theta}(\frac{1}{2} - \frac{1}{3} - \frac{1}{2}\theta^2 + \frac{1}{3}\theta^3) =$$
$$= \frac{2}{3}\theta^2 + \frac{2}{1-\theta}(\frac{1}{6} + \frac{2\theta^3 - 3\theta^2}{6}) = \frac{2\theta^2}{3} + \frac{2\theta^3 - 3\theta^2 + 1}{3(1-\theta)} = \frac{2\theta^2 - 2\theta^3 + 2\theta^3 - 3\theta^2 + 1}{3(1-\theta)} =$$
$$= \frac{1-\theta^2}{3(1-\theta)} = \underline{\frac{1+\theta}{3}}$$

Дисперсия

$$D\eta \stackrel{\text{def}}{=} M(\eta - M\eta)^2 = M\eta^2 - (M\eta)^2$$

$$M\eta^2 \stackrel{\text{def}}{=} \int_{\mathbb{R}} f(x)x^2dx = \int_0^\theta \frac{2x}{\theta}x^2dx + \int_\theta^1 \frac{2(1-x)}{1-\theta}x^2dx = \frac{2}{\theta}\int_0^\theta x^3dx + \frac{2}{1-\theta}\int_\theta^1 (x^2-x^3)dx =$$
$$= \frac{1}{2\theta}x^4\Big|_0^\theta + \frac{2}{1-\theta}(\frac{1}{3}x^3 - \frac{1}{4}x^4)\Big|_\theta^1 = \frac{1}{2\theta}\theta^4 + \frac{2}{1-\theta}(\frac{1}{3} - \frac{1}{4} - \frac{1}{3}\theta^3 + \frac{1}{4}\theta^4) =$$
$$= \frac{1}{2}\theta^3 + \frac{2}{1-\theta}(\frac{1}{12} + \frac{3\theta^4 - 4\theta^3}{12}) = \frac{1}{2}\theta^3 + \frac{1}{6(1-\theta)}(3\theta^4 - 4\theta^3 + 1) =$$
$$= \frac{1}{6(1-\theta)}(3\theta^4 - 4\theta^3 + 1 + 3\theta^3 - 3\theta^4) = \frac{1}{6(1-\theta)}(1-\theta^3) = \frac{1+\theta+\theta^2}{6}$$

$$\Rightarrow D\eta = M\eta^2 - (M\eta)^2 = \frac{1+\theta+\theta^2}{6} - (\frac{1+\theta}{3})^2 = \frac{3(1+\theta+\theta^2) - 2(1+2\theta+\theta^2)}{18} = \underline{\frac{1-\theta+\theta^2}{18}}$$

Квантиль уровня γ

$$F(x_\gamma) \geq \gamma \Rightarrow \begin{cases} x_\gamma = 0, & \text{если } \gamma < 0 \\ \dfrac{1}{\theta}x_\gamma^2 \geq \gamma, & \text{если } \gamma \in [0, \theta] \\ \dfrac{1}{1-\theta}(2x_\gamma - x_\gamma^2 - \theta) \geq \gamma, & \text{если } \gamma \in (\theta, 1] \\ x_\gamma = 1, & \text{если } \gamma > 1 \end{cases}$$

1) $\dfrac{1}{\theta}x_\gamma^2 \geq \gamma \Leftrightarrow x_\gamma \geq \sqrt{\theta\gamma} \Rightarrow x_\gamma = \sqrt{\theta\gamma}$

2) $\dfrac{1}{1-\theta}(2x_\gamma - x_\gamma^2 - \theta) \geq \gamma \Rightarrow \dfrac{1}{1-\theta}(2x_\gamma - x_\gamma^2 - \theta) = \gamma \Leftrightarrow -x_\gamma^2 + 2x_\gamma - \theta = (1-\theta)\gamma \Leftrightarrow$

$$\Leftrightarrow -x_\gamma^2 + 2x_\gamma - \theta - \gamma + \theta\gamma = 0 \Rightarrow$$
$$\Rightarrow D = 4 - 4(\theta + \gamma - \theta\gamma) = 4(1 - \theta - \gamma + \theta\gamma) \Rightarrow$$
$$\Rightarrow x_\gamma = \frac{-2 \pm 2\sqrt{1-\theta-\gamma+\theta\gamma}}{-2} = 1 \pm \sqrt{1-\theta-\gamma+\theta\gamma}\,.$$
$$x_\gamma \in [\theta, 1] \Rightarrow x_\gamma = 1 - \sqrt{1-\theta-\gamma+\theta\gamma}$$

$$\Rightarrow x_\gamma = \begin{cases} \sqrt{\theta\gamma}, & \text{если } \gamma \in [0, \theta] \\ 1 - \sqrt{1-\theta-\gamma+\theta\gamma}, & \text{если } \gamma \in (\theta, 1] \end{cases}$$

Задание 2

Треугольное распределение на практике используется часто, потому что оно имеет минимум, максимум и пик, что делает его уже достаточным к реальности распределением, так еще и оно очень простое по своей математике и применению. Конкретно в приведенной формуле распределение ограничено 0 и 1 и имеет пик в θ, а в обычных случаях оно позволяет посчитать предполагаемую прибыль какого-то ресторана, просто делая предположение о минимуме, максимуме и наиболее вероятном значении при помощи анализа полученного распределения (например через математическое ожидание). Также, в силу простоты, оно может служить некоторой заменой к другим распределениям подобной структуры. Так, если мы, например, наблюдаем образование бактерий на влажной сахарной линии, то очевидно, что надо использовать нормальное распределение, потому что это почти именно то, что оно и отображает. Однако, чтобы использовать нормальное распределение также практическим методом потребуется вычислить дисперсию, что может быть трудной задачей, потому временной заменой может послужить простое треугольное распределение, чтобы пронаблюдать на нем отклонения.

Задание 3

Чтобы построить выборку от равномерного случайного распределения требуется найти $F^{-1}(u)$, что мы фактически искали, вычисляя квантиль уровня γ. Чем я и воспользуюсь, описав код ниже.

In [2]:
```
import numpy as np
def generate_eta(theta=0.45):
    rng = np.random.default_rng()
    u = rng.uniform()
    if u <= theta: return (theta*u)**0.5
    return 1-(1-theta-u+theta*u)**0.5
```

Домашнее задание 2

Дискретное

Задание 1

In [3]:
```
# Здесь допустимо использование функций генераторов, указанных ранее
# theta задана в каждой функции генератора параметром по умолчанию
# потому отдельное упоминание не требуется
n = [5, 10, 100, 200, 400, 600, 800, 1000]
sample_xi = [[np.sort(np.array([generate_xi() for i in range(j)]))
              for i in range(5)] for j in n]

for i in range(len(n)):
    print('Пример сгенерированной выборки длины %d:'%n[i], end=' ')
    print(*sample_xi[i][0])
    print('-'*10)
```

Пример сгенерированной выборки длины 5: 2 6 6 23 26

Пример сгенерированной выборки длины 10: 3 3 6 9 14 18 21 24 27 28

Пример сгенерированной выборки длины 100: 1 1 1 2 2 2 2 2 2 3 3 3 3 4 4 4 6 6 6
6 6 7 7 8 8 8 9 9 10 10 10 10 11 11 11 11 11 12 13 13 13 13 13 13 13 14 14 14 14
15 15 15 15 15 15 15 15 15 16 17 17 17 18 18 18 18 19 20 20 20 20 21 21 21 21 23
23 23 24 24 24 25 25 25 25 25 25 26 26 26 26 26 26 27 27 28 28 28 28 29

Пример сгенерированной выборки длины 200: 1 1 1 1 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3
3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 5 5 5 5 5 5 5 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 9 9 9 10 10 10 10 10 10 10 10 10 11 11 11 12 12 12 12 12 12 12
12 12 12 12 12 12 13 13 13 13 14 14 14 15 15 15 15 15 15 15 15 15 16 16 16 17 17
17 17 17 17 18 18 18 18 18 19 19 20 20 20 20 20 20 20 20 21 21 21 21 21 22 22 22
22 22 22 22 22 22 22 22 23 23 23 23 24 24 24 24 24 24 24 24 24 24 24 24 25 25
25 25 25 25 25 26 26 26 27 27 27 27 27 27 28 28 28 28 28 28 28 29 29 29 29 29 29
29 29

Пример сгенерированной выборки длины 400: 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12 12
12 12 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 14 14 14
14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16 16 16
16 17 17 17 17 17 17 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18
18 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 21
21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22 22 22 22 23 23 23 23 23 23
23 23 23 23 23 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 25 25 25 25
25 25 25 25 25 25 25 25 25 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26
26 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 28 28 28 28 28 28 28 28 28 28
28 28 28 28 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29

Пример сгенерированной выборки длины 600: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5
5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7
7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 12
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 14 14 14
14 14 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 15 15 15
15 15 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17 17
17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21 21 21 21 21 21 22
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24
24 24 24 24 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 27 27
27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 28 28 28 28 28 28 28
28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 29 29 29 29 29 29 29 29
29 29 29 29 29 29 29 29 29 29 29 29 29

Пример сгенерированной выборки длины 800: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9

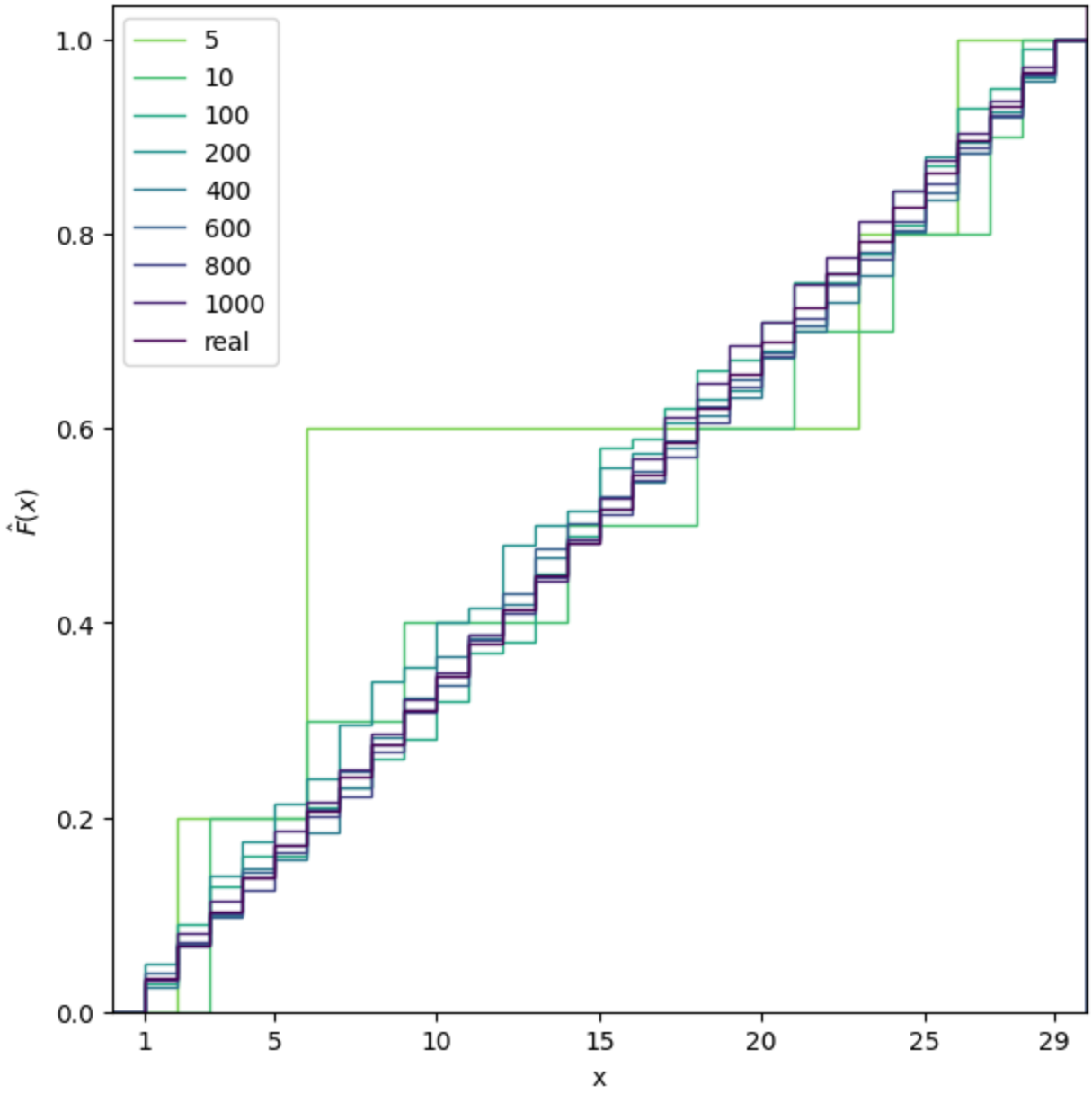
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 12 12 12
12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 14 14 14 14 14 14
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 15 15 15 15 15
15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 17 17 17 17 17 17
17 17 17 17 17 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 19 19 19 19 19
19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 25 25 25 25 25 25 25 25 25 25
25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 26 26 26 26 26 26
26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 27 27 27
27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 28 28 28
28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
28 28 28 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29
29 29 29 29 29

Пример сгенерированной выборки длины 1000: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 9 9 9 9
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 10 10 10 10 10 10 10 10 10 10 10 10
10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 11 11 11
11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
11 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12
13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13
13 13 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14
14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 15 15 15 15 15 15 15 15 15 15 15 15
15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
15 15 15 15 15 15 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16
16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 17 17 17 17 17 17 17
17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17 17
17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18
18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 19 19 19
19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19 19
19 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21
21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 21 21 21 21 21 21 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
22 22 22 22 22 22 22 22 22 22 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23
23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 23 24 24 24 24 24 24 24
24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 24 25 25
25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25 25
25 25 25 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26
26 26 26 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27
27 27 27 27 27 27 27 27 27 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28
28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 28 29 29 29 29 29 29 29 29 29 29
29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29 29

Задание 2

In [4]:
```
def xi_distr(sample, x):
    res = 0
    for i in sample:
        if i <= x:
            res += 1
    return res / len(sample)

def xi_distr_real(x: int, theta=29):
    return x / theta

X_realxi = np.arange(1-1, 29+1+1)
Y_realxi = xi_distr_real(X_realxi)

Yxi = np.array([[[xi_distr(sample_xi[k][j], x) for x in X_realxi]
                 for j in range(5)] for k in range(len(n))])

def xi_Dmn(Yn, Ym, n, m):
    res = 0
    for i in range(29):
        d = abs(Yn[i]-Ym[i])
        if d>res: res = d
    return (n*m/(n+m))**0.5*res

diffsxi = np.array([[[('%.2f' % xi_Dmn(Yxi[i][k], Yxi[j][k], n[i], n[i]))
                      if i>j else '-' for i in range(len(n))]
                     for j in range(len(n))]
                    for k in range(5)])
```

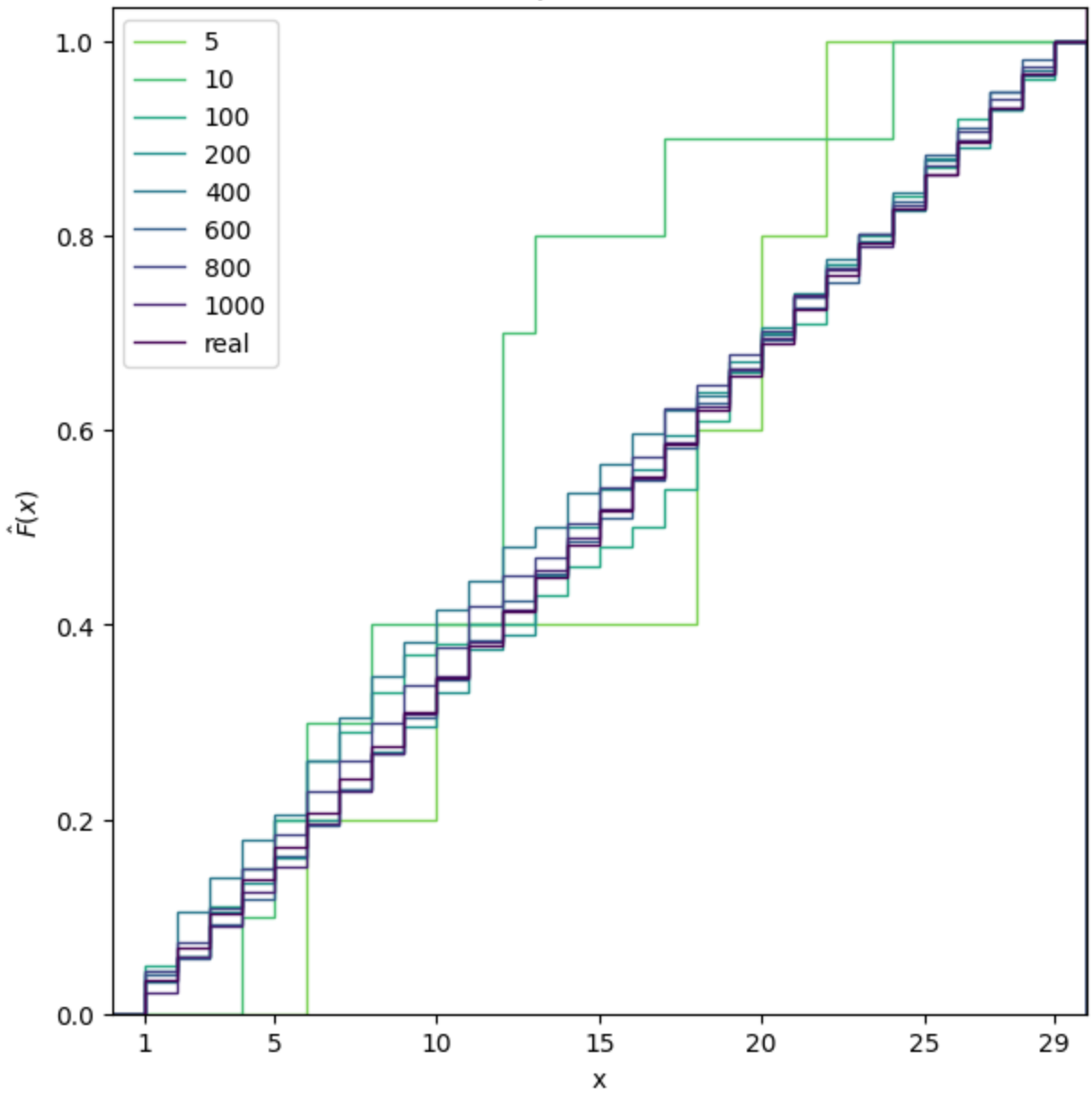
In [5]:
```
from matplotlib import pyplot as plt
colors = ['#7bd152', '#45be71', '#25a885', '#21908c',
          '#2b798e', '#355f8d', '#414486', '#482574']

# graph[0]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yxi[i][0], np.append(X_realxi, 30), color = colors[i])
ax[0].stairs(Y_realxi, np.append(X_realxi, 30), color = '#440154')
ax[0].set(xticks = [1]+list(range(5,26,5))+[29], xmargin = 0, ymargin = 0,
          xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Дискретное равномерное, выборка 1 \n$\\xi , \\,\\theta$ = 29')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[0], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$'+
                r'$|F_n(x)-F_m(x)|$');
```



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	2.76	3.60	5.87	6.78	7.97	8.56
10	-	-	0.92	1.05	1.63	1.76	1.98	2.50
100	-	-	-	1.00	0.88	0.87	1.37	1.16
200	-	-	-	-	0.92	1.13	1.47	1.48
400	-	-	-	-	-	0.49	0.45	1.22
600	-	-	-	-	-	-	0.61	0.98
800	-	-	-	-	-	-	-	0.92
1000	-	-	-	-	-	-	-	-

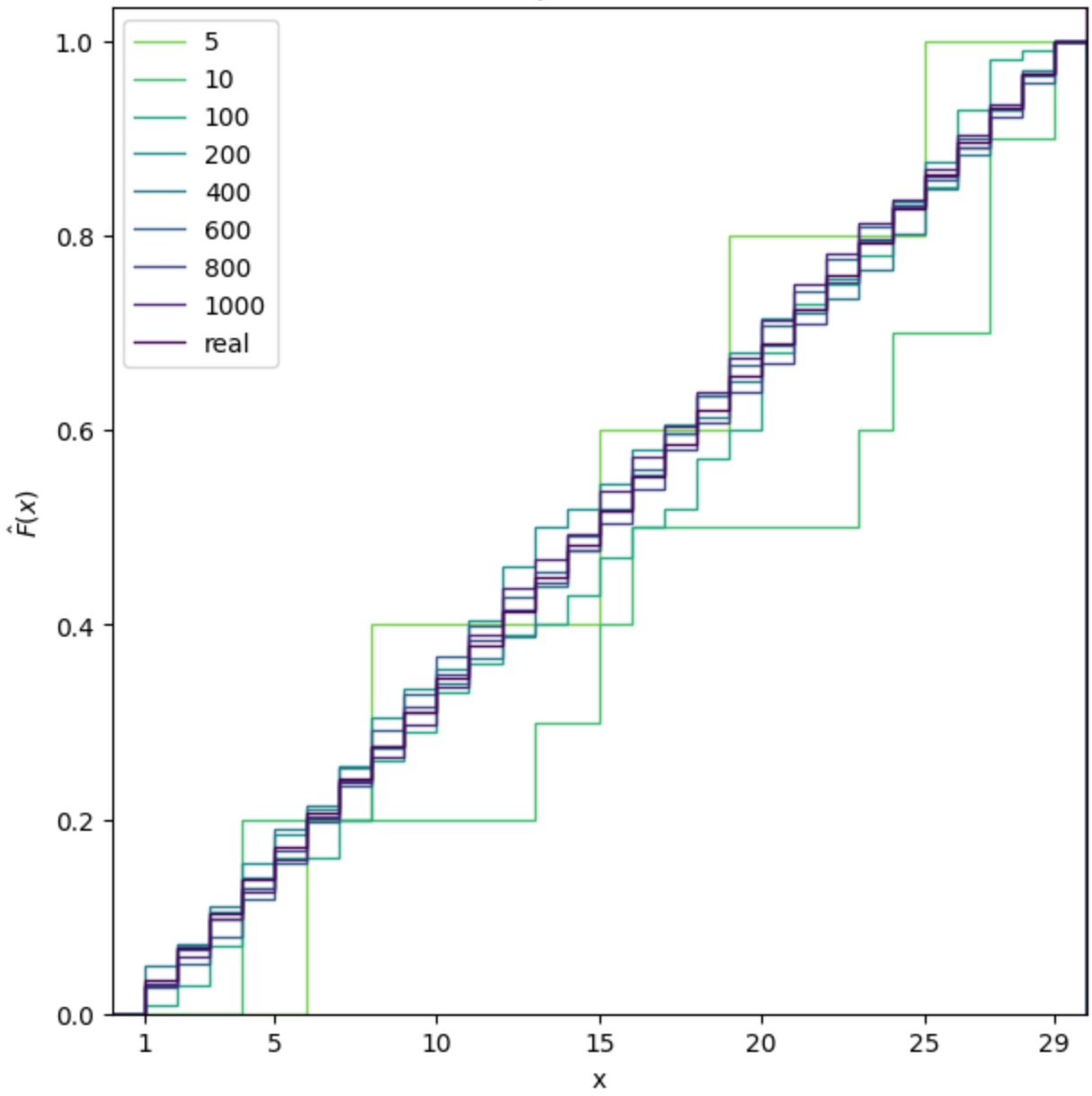
In [6]:
```
# graph[1]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yxi[i][1], np.append(X_realxi, 30), color = colors[i])
ax[0].stairs(Y_realxi, np.append(X_realxi, 30), color = '#440154')
ax[0].set(xticks = [1]+list(range(5,26,5))+[29], xmargin = 0, ymargin = 0,
          xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Дискретное равномерное, выборка 2 \n$\\xi , \\,\\theta$ = 29')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[1], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$'+
                r'$|F_n(x)-F_m(x)|$');
```



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	1.12	1.63	2.30	3.18	4.30	4.65	5.28
10	-	-	2.62	3.50	4.24	6.03	6.60	7.69
100	-	-	-	0.75	1.38	1.15	1.65	1.45
200	-	-	-	-	1.27	0.61	1.22	0.56
400	-	-	-	-	-	1.37	0.95	1.78
600	-	-	-	-	-	-	0.82	0.48
800	-	-	-	-	-	-	-	0.85
1000	-	-	-	-	-	-	-	-

In [7]:
```
# graph[2]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yxi[i][2], np.append(X_realxi, 30), color = colors[i])
ax[0].stairs(Y_realxi, np.append(X_realxi, 30), color = '#440154')
ax[0].set(xticks = [1]+list(range(5,26,5))+[29], xmargin = 0, ymargin = 0,
          xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Дискретное равномерное, выборка 3 \n$\\xi , \\,\\theta$ = 29')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[2], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$'+
                r'$|F_n(x)-F_m(x)|$');
```



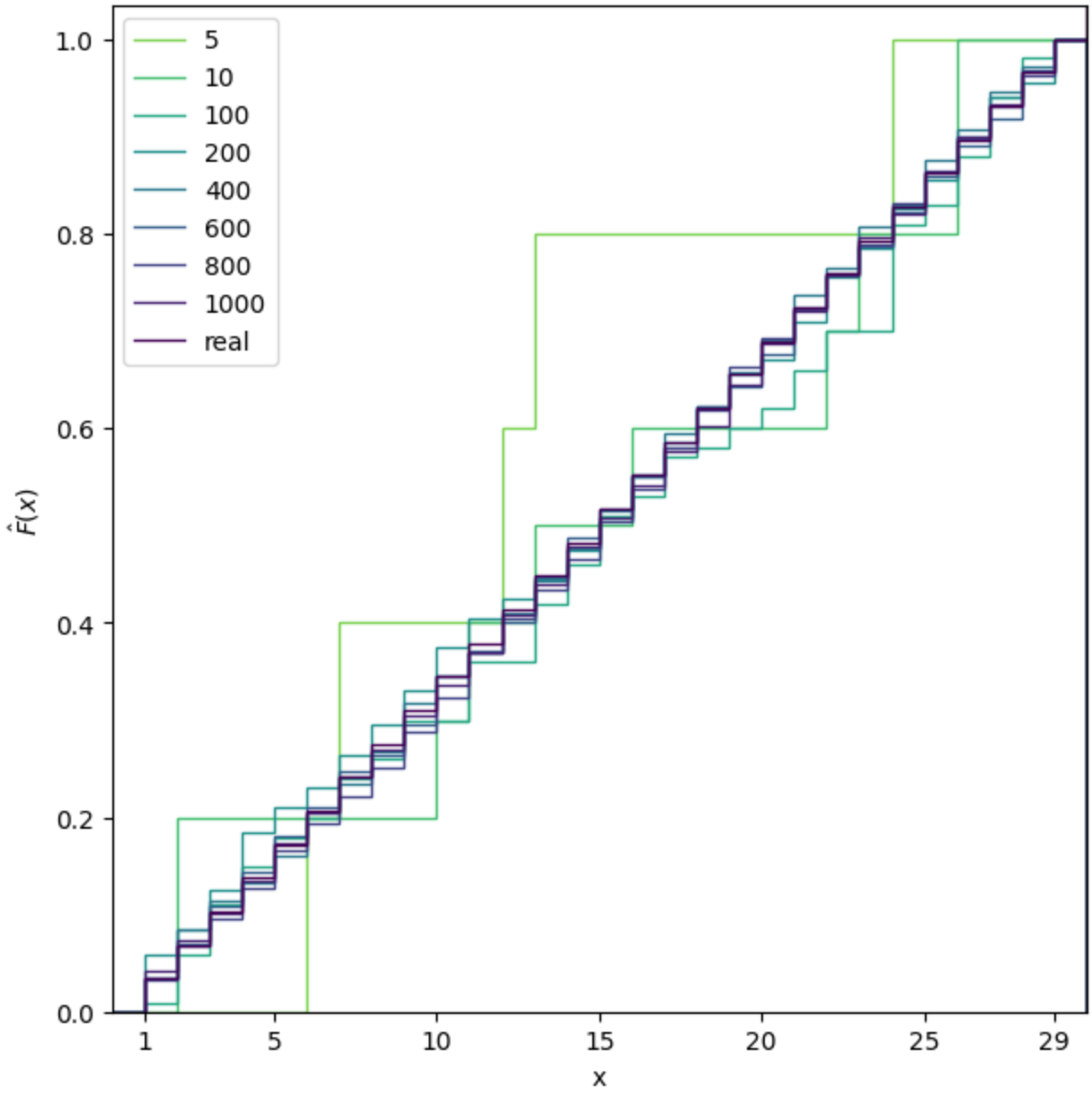
$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	1.41	1.85	2.69	2.68	3.38	3.53
10	-	-	1.77	2.60	3.32	4.79	5.05	6.31
100	-	-	-	1.00	1.10	1.33	1.20	1.88
200	-	-	-	-	1.03	0.78	1.13	0.92
400	-	-	-	-	-	0.75	0.62	1.13
600	-	-	-	-	-	-	0.79	0.70
800	-	-	-	-	-	-	-	0.99
1000	-	-	-	-	-	-	-	-

In [8]:
```
# graph[3]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yxi[i][3], np.append(X_realxi, 30), color = colors[i])
ax[0].stairs(Y_realxi, np.append(X_realxi, 30), color = '#440154')
ax[0].set(xticks = [1]+list(range(5,26,5))+[29], xmargin = 0, ymargin = 0,
          xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Дискретное равномерное, выборка 4 \n$\\xi , \\,\\theta$ = 29')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[3], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$'+
                r'$|F_n(x)-F_m(x)|$');
```

Дискретное равномерное, выборка 4

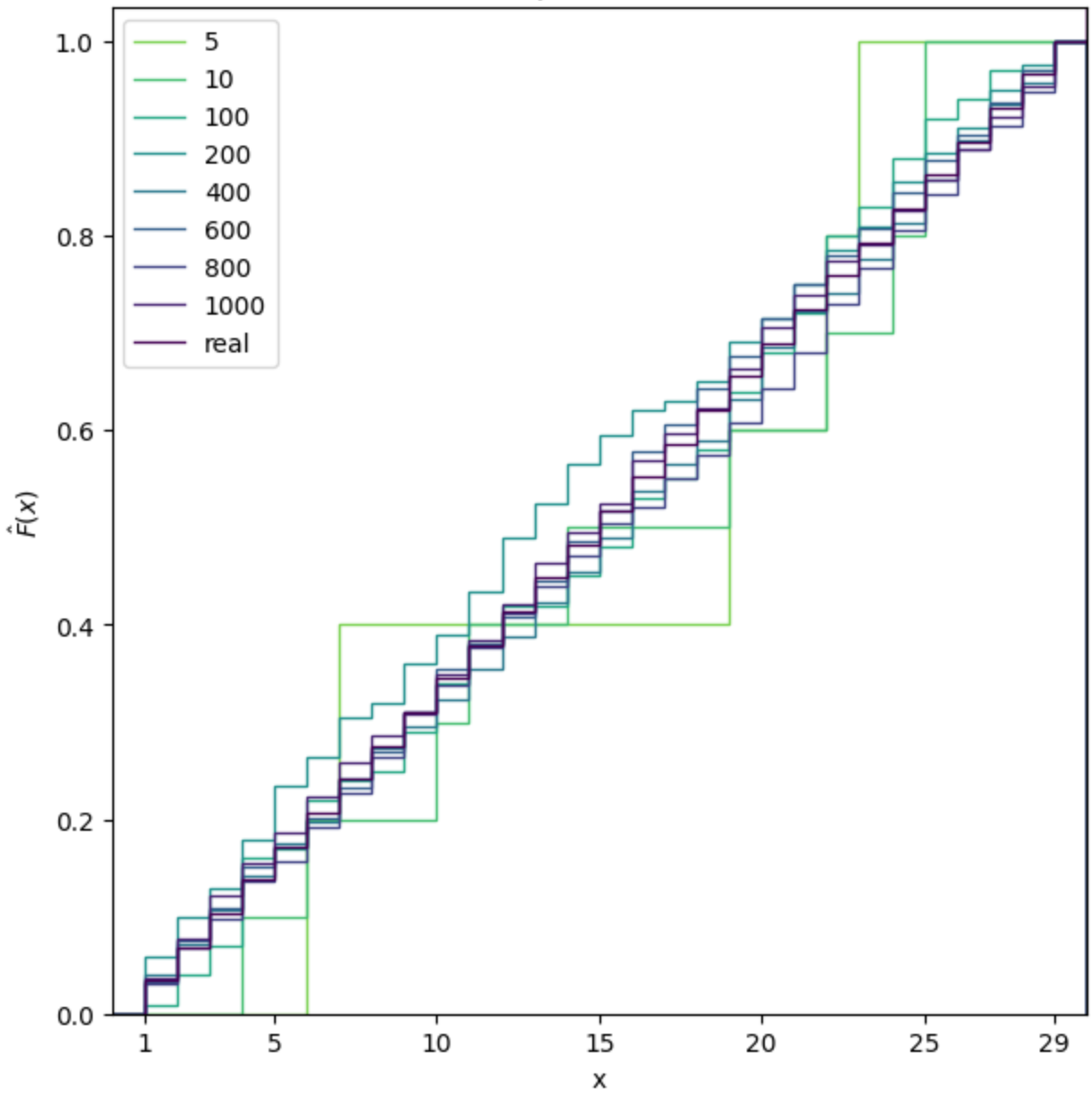
ξ, $\theta = 29$



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	2.69	3.55	5.06	6.12	7.33	8.05
10	-	-	0.99	1.30	1.94	2.25	2.65	2.84
100	-	-	-	0.85	1.52	1.50	1.77	2.15
200	-	-	-	-	0.74	0.69	1.15	1.12
400	-	-	-	-	-	0.46	0.57	0.44
600	-	-	-	-	-	-	0.48	0.45
800	-	-	-	-	-	-	-	0.44
1000	-	-	-	-	-	-	-	-

In [9]:
```
# graph[4]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yxi[i][4], np.append(X_realxi, 30), color = colors[i])
ax[0].stairs(Y_realxi, np.append(X_realxi, 30), color = '#440154')
ax[0].set(xticks = [1]+list(range(5,26,5))+[29], xmargin = 0, ymargin = 0,
          xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Дискретное равномерное, выборка 5 \n$\\xi , \\,\\theta$ = 29')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[4], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$'+
                r'$|F_n(x)-F_m(x)|$');
```



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	1.27	2.50	3.18	4.21	4.68	4.99
10	-	-	0.92	1.60	2.02	2.60	3.15	3.20
100	-	-	-	1.15	0.95	1.10	1.55	1.41
200	-	-	-	-	1.56	1.41	1.98	1.57
400	-	-	-	-	-	0.92	0.85	0.91
600	-	-	-	-	-	-	1.45	0.57
800	-	-	-	-	-	-	-	1.40
1000	-	-	-	-	-	-	-	-

Задание 3

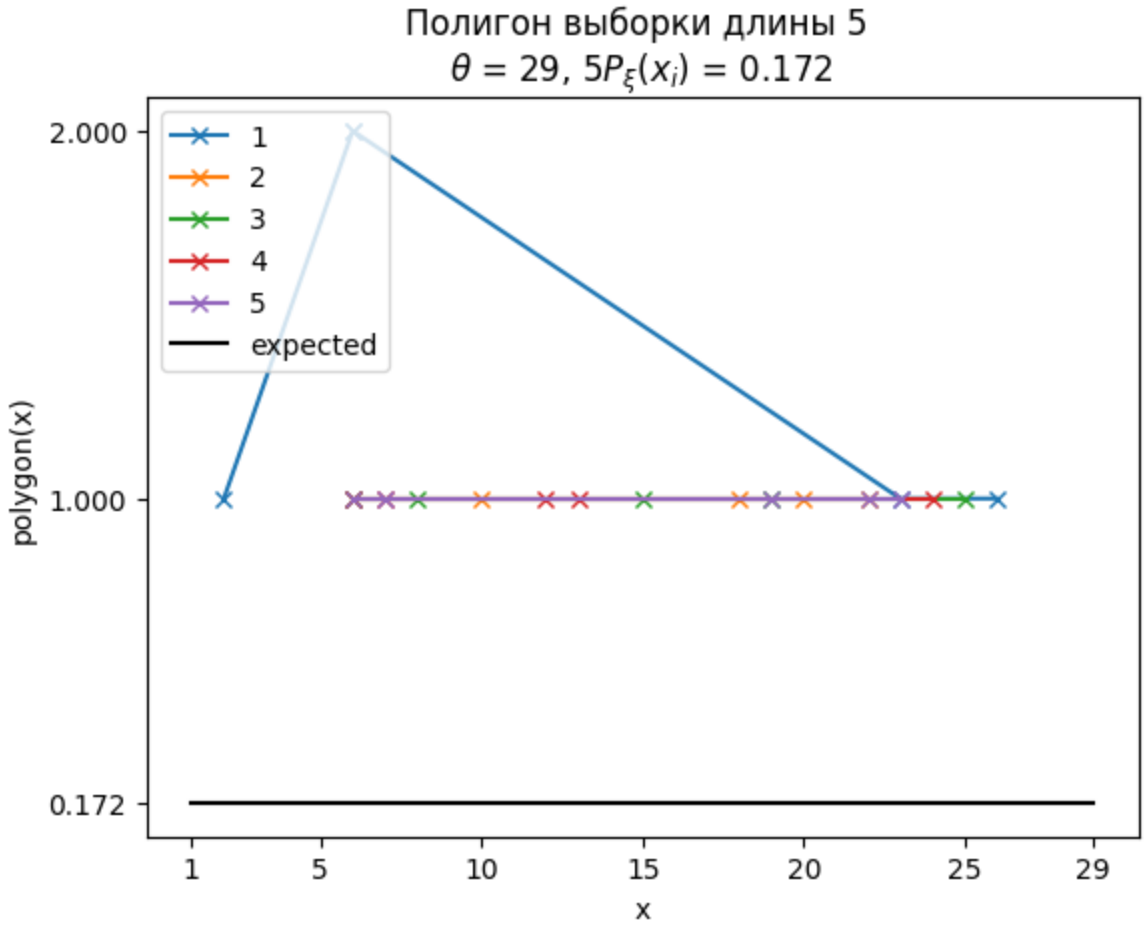
Чтобы сравнить график полигона частот и график функции вероятности нужно в идеале изобразить один на другом, при этом как-то правильно их друг с другом соотнести, чтобы они правильно наложились друг на друга. Здесь я попробую подвести график вероятности к графику полигона частот (то есть второй останется неизменным, а первый будет домножен). Чтобы вычислить коэффициент домножения рассмотрим $\hat{F}(x)$:

$\hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} I(x_i \leq x) \Rightarrow \hat{P}(x_i) = \hat{F}(x_i) - \hat{F}(x_{i-1}) = \frac{1}{n} k\, I(x_i = x_i) = \frac{k}{n}$, где k - частота встречаемости элемента в выборке.

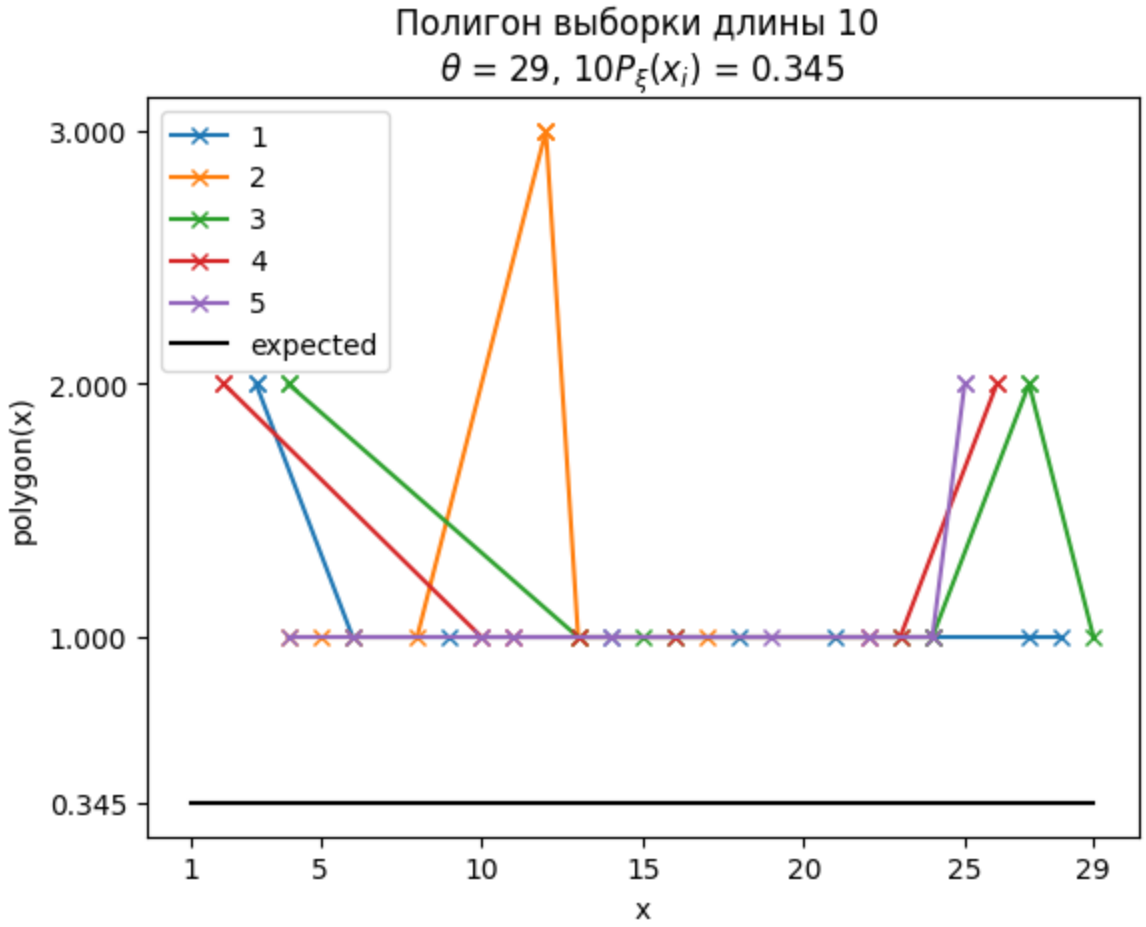
$\Rightarrow$ при $n \to \infty$ $\hat{P}(x_i) \to P(x_i) \Rightarrow P(x_i) = \frac{k}{n} \Rightarrow k = nP(x_i)$ при $n \to \infty$, что значит, что график вероятности подводим к полигону частот через домножение на количество элементов в выборке.

In [10]:
```
def xi_pilygon(sample, x):
    return np.count_nonzero(sample==x)

def min_t(samples):
    res = samples[0][0]
    for i in samples:
        t = min(i)
        if t < res: res = t
    return res

def max_t(samples):
    res = samples[0][0]
    for i in samples:
        t = max(i)
        if t > res: res = t
    return res

Y_polxi = [[np.array([xi_pilygon(sample_xi[k][j], sample_xi[k][j][i])
                      for i in range(n[k])])
           for j in range(5)] for k in range(len(n))]

y_limsxi = np.array([[min_t(j), max_t(j)] for j in Y_polxi])

posibilityxi = np.full((1000), 1/29)
```

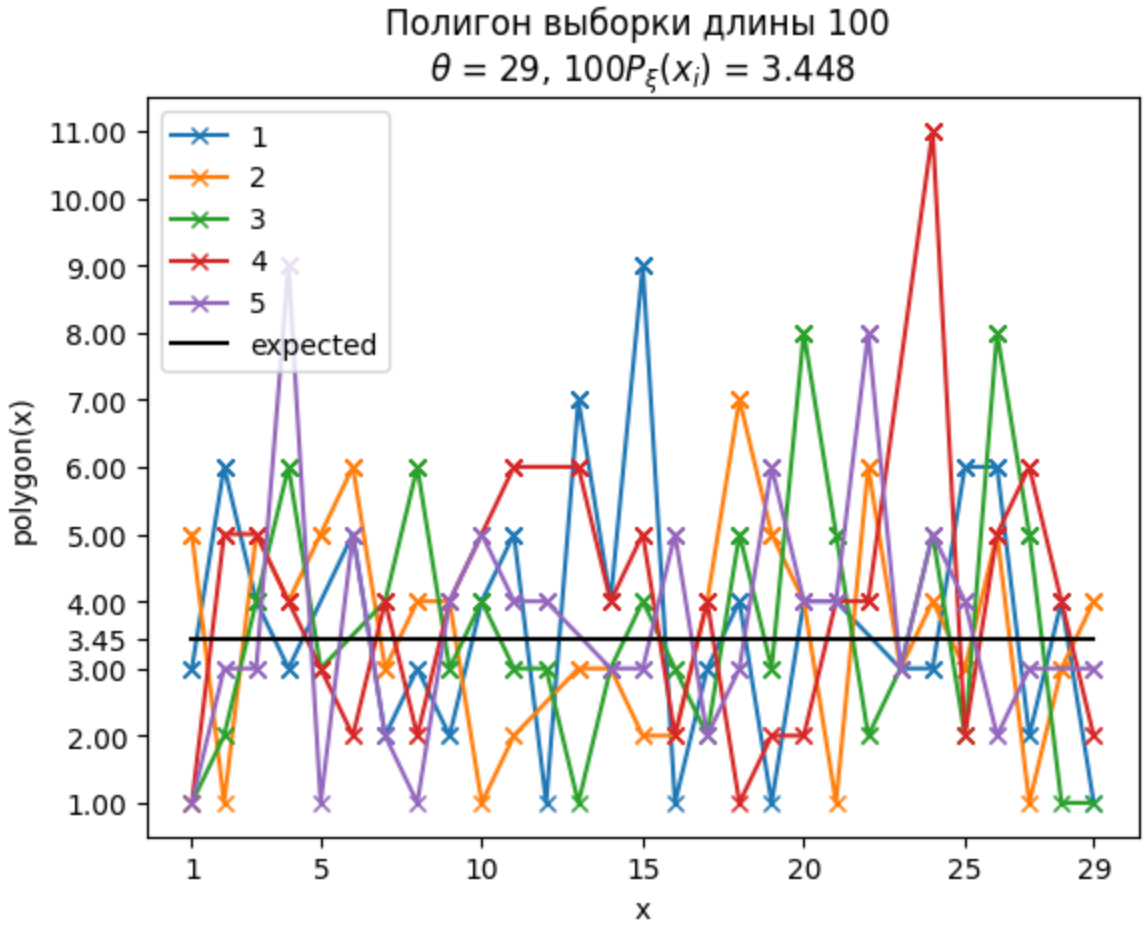
In [11]:
```
# n=5
for i in range(5):
    plt.plot(sample_xi[0][i], Y_polxi[0][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*5, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.append(np.arange(y_limsxi[0,0], y_limsxi[0,1]+1), 5/29))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 5 \n' +
          '$\\theta$ = 29, $5P_{\\xi}(x_i)$ = %.3f'% (5/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



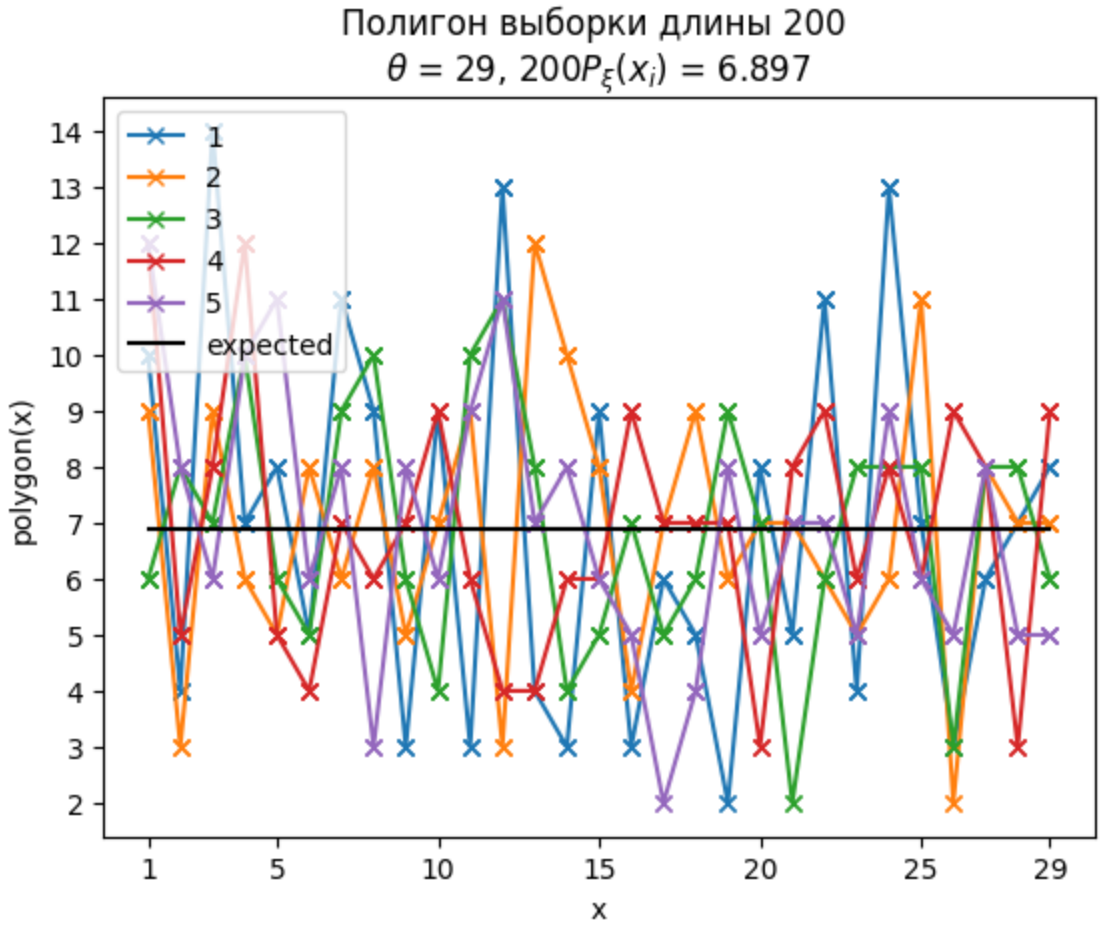
In [12]:
```
# n=10
for i in range(5):
    plt.plot(sample_xi[1][i], Y_polxi[1][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*10, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.append(np.arange(y_limsxi[1,0], y_limsxi[1,1]+1), 10/29))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 10 \n' +
          '$\\theta$ = 29, $10P_{\\xi}(x_i)$ = %.3f'% (10/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



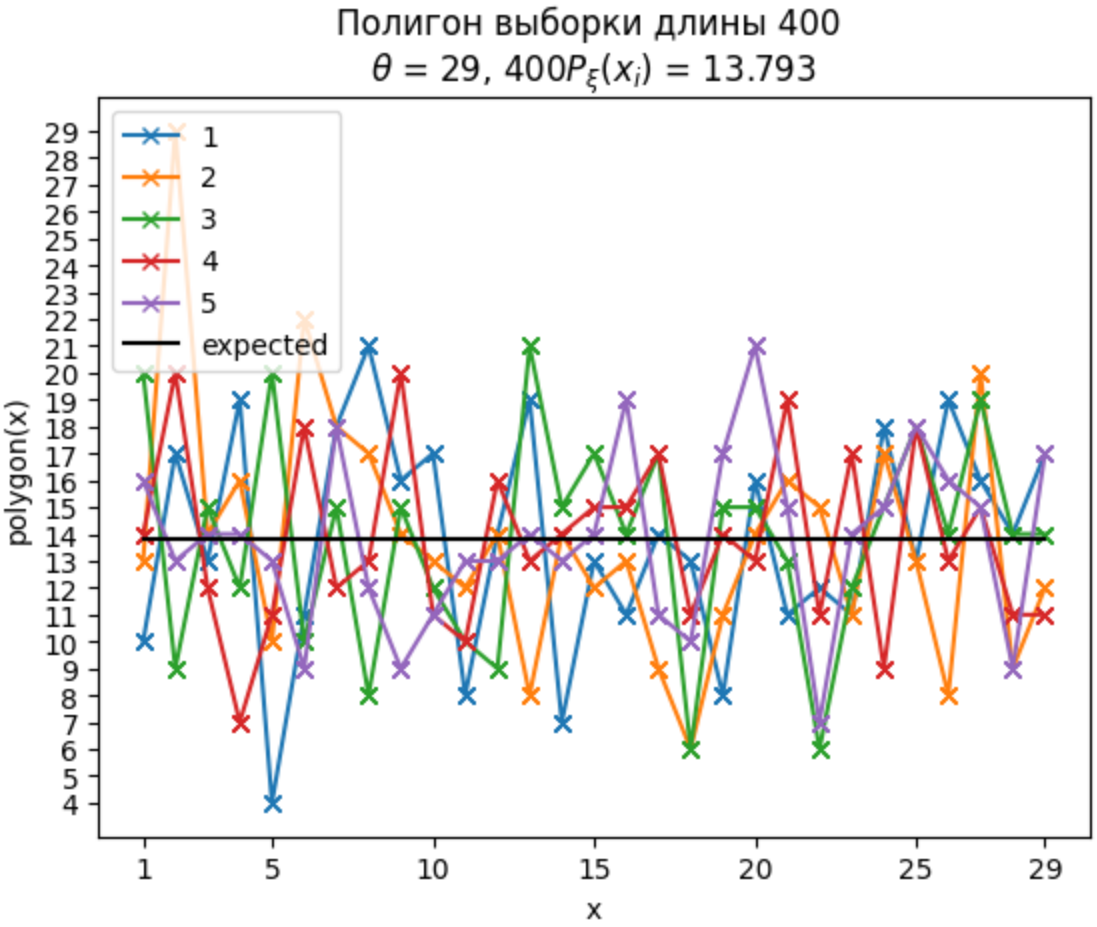
In [13]:
```
# n=100
for i in range(5):
    plt.plot(sample_xi[2][i], Y_polxi[2][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*100, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.append(np.arange(y_limsxi[2,0], y_limsxi[2,1]+1), 100/29))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 100 \n' +
          '$\\theta$ = 29, $100P_{\\xi}(x_i)$ = %.3f'% (100/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



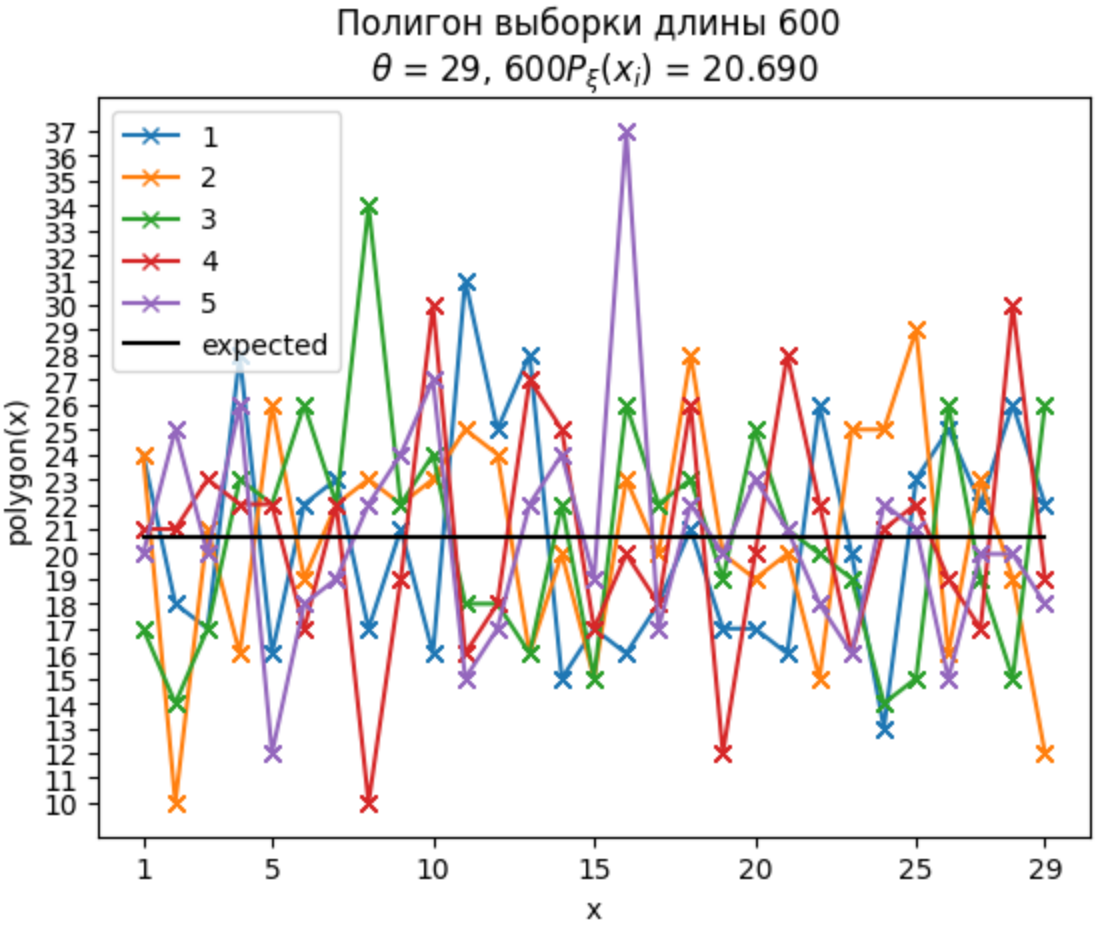
In [14]:
```
# n=200
for i in range(5):
    plt.plot(sample_xi[3][i], Y_polxi[3][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*200, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.arange(y_limsxi[3,0], y_limsxi[3,1]+1))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 200 \n' +
          '$\\theta$ = 29, $200P_{\\xi}(x_i)$ = %.3f'% (200/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



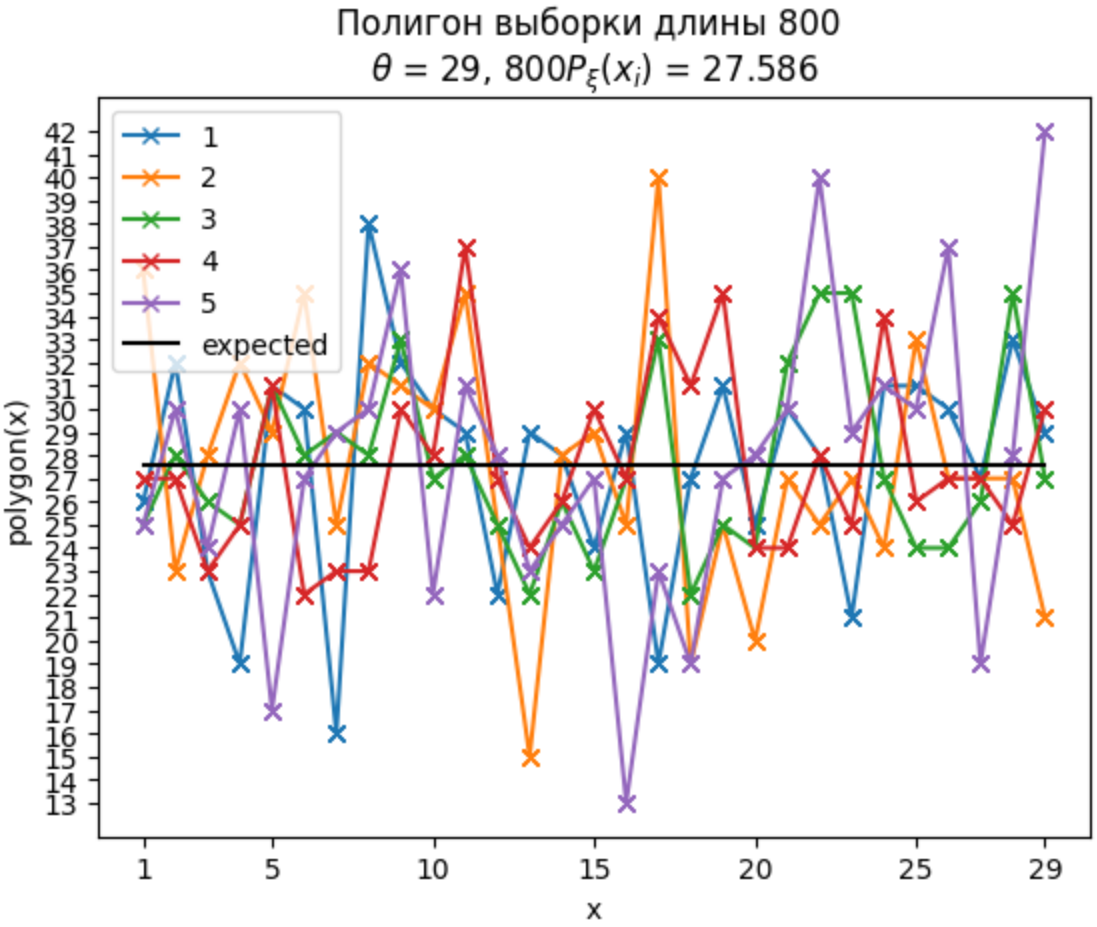
In [15]:
```
# n=400
for i in range(5):
    plt.plot(sample_xi[4][i], Y_polxi[4][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*400, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.arange(y_limsxi[4,0], y_limsxi[4,1]+1))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 400 \n' +
          '$\\theta$ = 29, $400P_{\\xi}(x_i)$ = %.3f'% (400/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



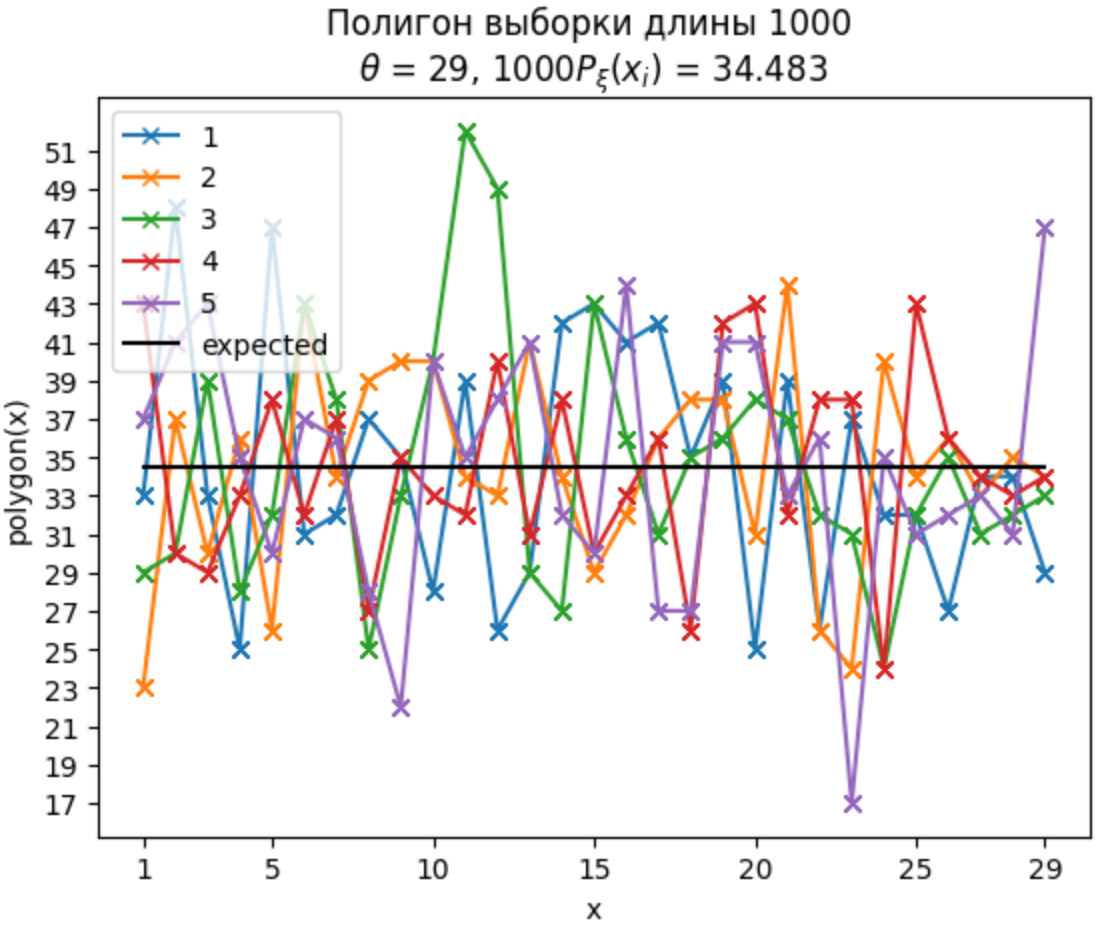
In [16]:
```
# n=600
for i in range(5):
    plt.plot(sample_xi[5][i], Y_polxi[5][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*600, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.arange(y_limsxi[5,0], y_limsxi[5,1]+1))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 600 \n' +
          '$\\theta$ = 29, $600P_{\\xi}(x_i)$ = %.3f'% (600/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



In [17]:
```
# n=800
for i in range(5):
    plt.plot(sample_xi[6][i], Y_polxi[6][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*800, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.arange(y_limsxi[6,0], y_limsxi[6,1]+1))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 800 \n' +
          '$\\theta$ = 29, $800P_{\\xi}(x_i)$ = %.3f'% (800/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



In [18]:
```
# n=1000
for i in range(5):
    plt.plot(sample_xi[7][i], Y_polxi[7][i], '-x')
plt.plot(np.linspace(1, 29, 1000), posibilityxi*1000, 'k')
plt.xticks([1]+list(range(5,26,5))+[29])
plt.yticks(np.arange(y_limsxi[7,0], y_limsxi[7,1]+1, 2))
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 1000 \n' +
          '$\\theta$ = 29, $1000P_{\\xi}(x_i)$ = %.3f'% (1000/29))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



По полигону частот и графику домноженной вероятности можно заметить, что встречаемость возможных значений распределения держится вокруг вероятности каждого из них, и с увеличением числа элементов выборки эту закономерность наблюдать становится легче. Это наблюдение соответствует теореме о схождении эмперической вероятности к математической (в курсе лекций это теорема о функциях распределения, но вероятность из этого следует). Конечно, у нас мог произойти случай, когда все элементы выборки попали в 1, но шанс этого в силу построения равен 29^{-1000}, так что дополнительный разброс в виде 5 выборок на каждое указанное количество элементов создает общую картину, которая со стремлением n к бесконечности, устремит полигон частот к домноженному графику вероятности. Также следует заметить, что на графиках в 5 и 10 элементов выборки домноженная вероятность даже не близка к частотам. Это, во-первых, еще раз подтверждает теорему, а во-вторых, домноженная вероятность сильно меньше единицы, что еще больше мешает приближению.

Задание 4

In [19]:
```
def xi_sample_mean(sample):
    return sum(sample)/len(sample)

def xi_sample_variance(sample):
    return sum((sample-xi_sample_mean(sample))**2)/len(sample)

meansxi = np.array([[xi_sample_mean(sample_xi[k][j])for j in range(5)]
                    for k in range(len(n))])
variancesxi = np.array([[xi_sample_variance(sample_xi[k][j])for j in range(5)]
                        for k in range(len(n))])
means_pxi = np.array([['%.2f' % i for i in j] for j in meansxi])
variances_pxi = np.array([['%.2f' % i for i in j] for j in variancesxi])
expectationxi = (29+1)/2
variancexi = (29*29-1)/12
means_difxi = np.array([['%.2f' % i for i in j] for j in (meansxi-expectationxi)])
variances_difxi = np.array([['%.2f' % i for i in j]
                            for j in (variancesxi-variancexi)])
```

Для начала следует продемонстрировать выборочные средние и выборочные дисперсии, чтобы дать большее придставление о числах, с которыми идет работа. Благо их не так много.

In [20]:
```
fig, ax = plt.subplots(2,1, figsize=(7,6.5))
ax[0].table(cellText = means_pxi, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[0].set_axis_off()
ax[0].set_title('Выборочные средние');
ax[1].table(cellText = variances_pxi, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title('Выборочные дисперсии');
```

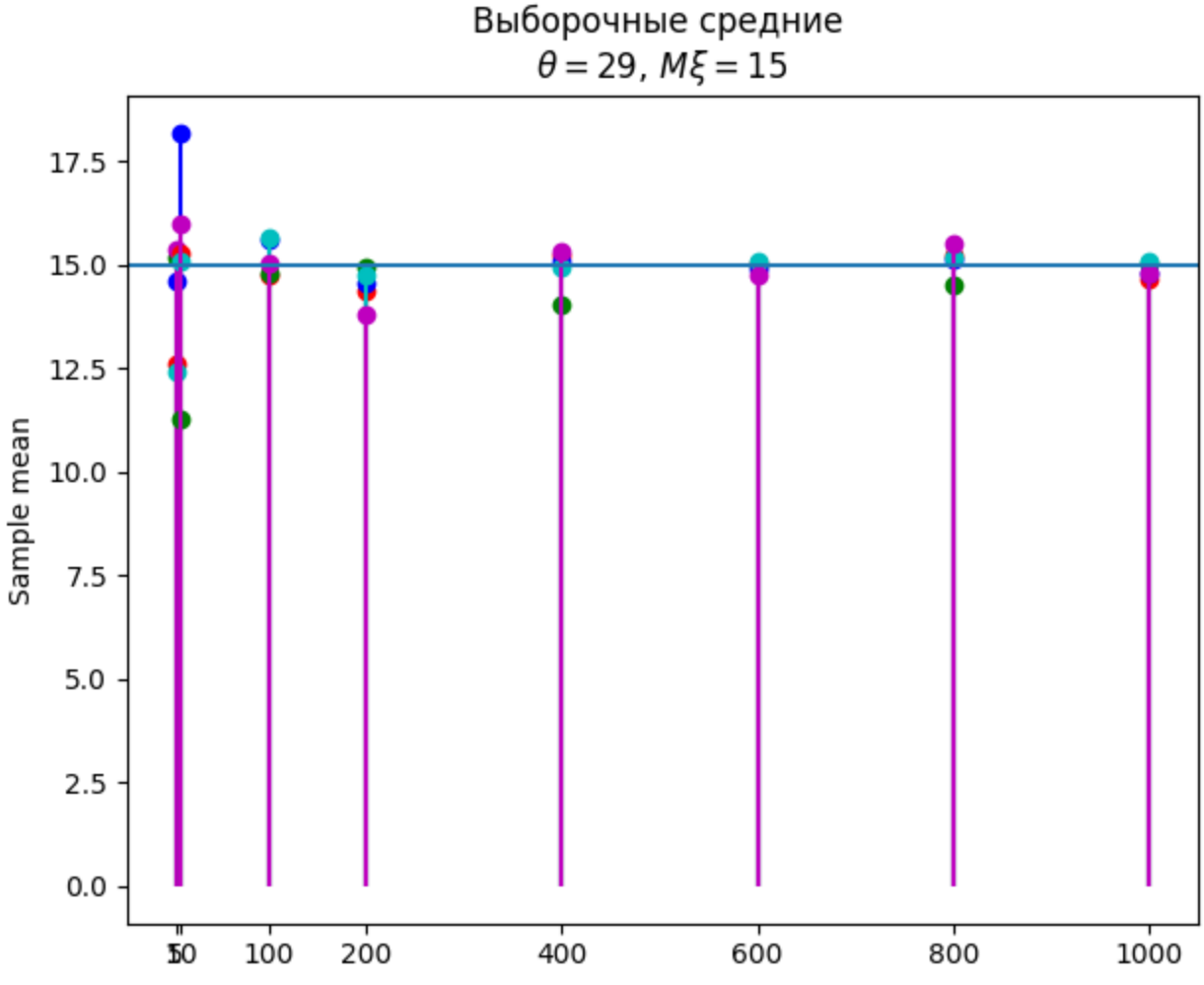
Выборочные средние

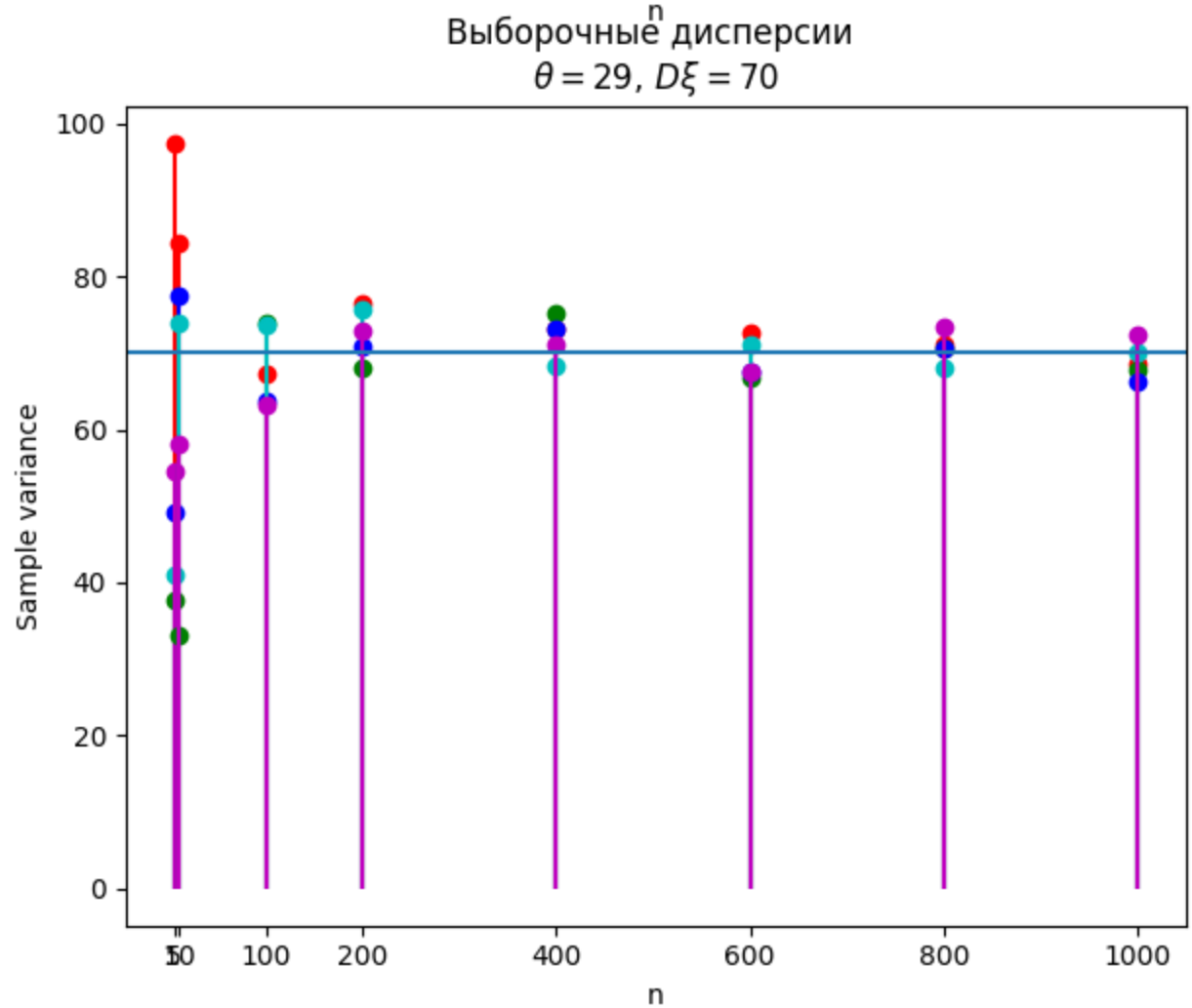
	1	2	3	4	5
5	12.60	15.20	14.60	12.40	15.40
10	15.30	11.30	18.20	15.10	16.00
100	14.78	14.82	15.60	15.68	15.05
200	14.37	14.95	14.57	14.78	13.82
400	15.21	14.04	15.11	14.93	15.31
600	15.03	14.99	14.88	15.07	14.77
800	15.21	14.50	15.13	15.21	15.51
1000	14.68	15.04	14.81	15.10	14.80

Выборочные дисперсии

	1	2	3	4	5
5	97.44	37.76	49.04	41.04	54.64
10	84.41	33.01	77.36	73.89	58.00
100	67.39	73.87	63.72	73.60	63.09
200	76.48	67.93	70.80	75.74	72.95
400	73.14	75.08	73.07	68.36	71.15
600	72.71	66.73	67.63	71.11	67.47
800	71.07	70.68	70.67	68.04	73.46
1000	68.51	67.71	66.27	70.20	72.26

In [21]:
```
fig, ax = plt.subplots(2,1, figsize=(7,12))
for k in range(len(n)):
    for j in range(5):
        ax[0].stem(n[k], meansxi[k][j], 'rgbcm'[j])
ax[0].axhline(expectationxi)
ax[0].set(xticks = n, xlabel = 'n', ylabel = 'Sample mean',
          title = 'Выборочные средние \n' +
                  '$\\theta = 29, \\,M\\xi = %d$'%expectationxi)

for k in range(len(n)):
    for j in range(5):
        ax[1].stem(n[k], variancesxi[k][j], 'rgbcm'[j])
ax[1].axhline(y = variancexi)
ax[1].set(xticks = n, xlabel = 'n', ylabel = 'Sample variance',
          title = 'Выборочные дисперсии \n' +
                  '$\\theta = 29, \\,D\\xi = %d$'%variancexi);
```


Выборочные средние
$\theta = 29$, $M\xi = 15$
Sample mean
n
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
100
200
400
600
800
1000


Выборочные дисперсии
$\theta = 29$, $D\xi = 70$
Sample variance
n
100
80
60
40
20
0
100
200
400
600
800
1000

По графикам можно заметить, что с увеличением количества элементов выборки и математическое ожидание, и дисперсия сходятся к предполагаемому значению. Что еще раз подтверждает теорему, упомянутую в предыдущей задаче.

In [22]:
```
fig, ax = plt.subplots(2,1, figsize=(7,7))
ax[0].table(cellText = means_difxi, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[0].set_axis_off()
ax[0].set_title('Смещение оценки: $b(\\theta)=M_{\\theta}T(x)-\\tau(\\theta)$' +
                '\n $M\\xi = $'+str(expectationxi));

ax[1].table(cellText = variances_difxi, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title('Разница выборочной дисперсии и дисперсии' +
                '\n $D\\xi = $'+str(variancexi));
```

Смещение оценки: $b(\theta) = M_\theta T(x) - \tau(\theta)$
$M\xi = 15.0$

	1	2	3	4	5
5	-2.40	0.20	-0.40	-2.60	0.40
10	0.30	-3.70	3.20	0.10	1.00
100	-0.22	-0.18	0.60	0.68	0.05
200	-0.63	-0.05	-0.43	-0.22	-1.18
400	0.21	-0.96	0.11	-0.07	0.31
600	0.03	-0.01	-0.12	0.07	-0.23
800	0.21	-0.50	0.13	0.21	0.51
1000	-0.32	0.04	-0.19	0.10	-0.20

Разница выборочной дисперсии и дисперсии
$D\xi = 70.0$

	1	2	3	4	5
5	27.44	-32.24	-20.96	-28.96	-15.36
10	14.41	-36.99	7.36	3.89	-12.00
100	-2.61	3.87	-6.28	3.60	-6.91
200	6.48	-2.07	0.80	5.74	2.95
400	3.14	5.08	3.07	-1.64	1.15
600	2.71	-3.27	-2.37	1.11	-2.53
800	1.07	0.68	0.67	-1.96	3.46
1000	-1.49	-2.29	-3.73	0.20	2.26

По таблице смещения оценок можно заметить, что с увеличением числа элементов выборки, модуль смещения оценки уменьшается, что свидетельствует о состоятельности оценки $\frac{1}{n}\sum_{i=1}^n I(x

Также эти таблицы (как и предыдущее много чего) еще раз демонстрируют справедливость теоремы о схождении функций распределения.

Абсолютно непрерывное

Большая часть материала уже разобрана выше на примере дискретного случая. Поэтому здесь различных описаний будет меньше, так как они будут почти 1 в 1 повторяться. Это самое почти, при необходимости, и будет описываться.

Задание 1

In [23]:
```
sample_eta = [[np.sort(np.array([generate_eta() for i in range(j)]))
               for i in range(5)] for j in n]

#demo_peta = np.array(['%.3f'%i for i in sample_eta[7][0]]).reshape((-1, 20))
for i in range(len(n)):
    demo_peta = ['%.3f'%j for j in sample_eta[i][0]]
    print('Пример сгенерированной выборки длины %d:'%n[i], end=' ')
    print(*demo_peta)
    print('-'*10)
```

Пример сгенерированной выборки длины 5: 0.404 0.408 0.699 0.780 0.806

Пример сгенерированной выборки длины 10: 0.156 0.274 0.366 0.421 0.528 0.532 0.5
69 0.666 0.707 0.868

Пример сгенерированной выборки длины 100: 0.092 0.133 0.152 0.175 0.183 0.192 0.
203 0.226 0.255 0.263 0.268 0.275 0.277 0.278 0.280 0.286 0.309 0.315 0.316 0.31
8 0.321 0.325 0.334 0.337 0.366 0.367 0.367 0.373 0.381 0.384 0.384 0.386 0.389
0.394 0.407 0.413 0.416 0.417 0.420 0.424 0.428 0.435 0.444 0.453 0.460 0.466 0.
466 0.470 0.472 0.474 0.477 0.477 0.482 0.513 0.518 0.519 0.521 0.522 0.526 0.53
8 0.539 0.541 0.545 0.547 0.556 0.574 0.588 0.589 0.596 0.599 0.602 0.602 0.602
0.616 0.648 0.653 0.654 0.661 0.661 0.678 0.681 0.682 0.686 0.686 0.687 0.692 0.
692 0.716 0.733 0.743 0.744 0.746 0.762 0.766 0.805 0.825 0.874 0.886 0.902 0.92
9

Пример сгенерированной выборки длины 200: 0.030 0.046 0.103 0.106 0.128 0.134 0.
134 0.138 0.152 0.154 0.156 0.176 0.181 0.183 0.183 0.184 0.190 0.200 0.200 0.20
4 0.204 0.208 0.219 0.222 0.230 0.230 0.231 0.232 0.237 0.237 0.245 0.245 0.246
0.246 0.247 0.254 0.257 0.266 0.286 0.294 0.297 0.298 0.308 0.314 0.315 0.316 0.
317 0.324 0.327 0.327 0.330 0.336 0.336 0.346 0.347 0.352 0.355 0.361 0.362 0.36
7 0.371 0.371 0.379 0.382 0.384 0.387 0.387 0.387 0.395 0.395 0.398 0.404 0.406
0.410 0.414 0.416 0.418 0.425 0.425 0.426 0.428 0.431 0.433 0.437 0.438 0.438 0.
440 0.441 0.443 0.444 0.446 0.446 0.447 0.452 0.454 0.455 0.456 0.456 0.470 0.47
2 0.473 0.475 0.475 0.475 0.477 0.478 0.483 0.486 0.490 0.495 0.500 0.502 0.506
0.508 0.511 0.517 0.517 0.525 0.526 0.528 0.529 0.529 0.535 0.540 0.549 0.558 0.
562 0.566 0.568 0.572 0.583 0.592 0.595 0.599 0.601 0.602 0.612 0.617 0.624 0.62
5 0.625 0.626 0.629 0.635 0.636 0.640 0.643 0.648 0.652 0.653 0.670 0.671 0.682
0.684 0.690 0.693 0.694 0.694 0.696 0.700 0.706 0.713 0.714 0.721 0.728 0.730 0.
742 0.751 0.762 0.764 0.773 0.774 0.776 0.779 0.782 0.785 0.795 0.810 0.811 0.81
1 0.813 0.817 0.820 0.821 0.821 0.828 0.829 0.830 0.831 0.832 0.842 0.852 0.853
0.855 0.869 0.934 0.955 0.966 0.986 0.987

Пример сгенерированной выборки длины 400: 0.032 0.047 0.050 0.056 0.073 0.091 0.
096 0.105 0.110 0.110 0.112 0.114 0.118 0.122 0.123 0.124 0.126 0.130 0.139 0.14
3 0.149 0.153 0.157 0.157 0.157 0.161 0.165 0.173 0.174 0.182 0.184 0.191 0.194
0.194 0.195 0.205 0.206 0.208 0.209 0.211 0.221 0.222 0.225 0.228 0.230 0.231 0.
232 0.241 0.246 0.246 0.246 0.248 0.250 0.253 0.253 0.255 0.258 0.260 0.262 0.26
4 0.268 0.270 0.275 0.277 0.277 0.284 0.285 0.287 0.287 0.294 0.295 0.295 0.296
0.298 0.298 0.301 0.305 0.308 0.309 0.310 0.312 0.312 0.313 0.318 0.319 0.320 0.
320 0.321 0.322 0.329 0.329 0.330 0.330 0.332 0.333 0.335 0.337 0.338 0.339 0.33
9 0.339 0.341 0.341 0.343 0.343 0.344 0.346 0.347 0.348 0.355 0.358 0.358 0.361
0.363 0.363 0.363 0.365 0.367 0.367 0.369 0.369 0.375 0.381 0.382 0.383 0.384 0.
385 0.388 0.388 0.388 0.391 0.394 0.394 0.395 0.396 0.400 0.400 0.400 0.400 0.40
0 0.401 0.401 0.402 0.404 0.404 0.406 0.407 0.408 0.409 0.409 0.409 0.409 0.410
0.412 0.412 0.413 0.413 0.415 0.416 0.418 0.418 0.418 0.419 0.421 0.423 0.424 0.
424 0.426 0.426 0.430 0.431 0.432 0.436 0.438 0.438 0.440 0.444 0.444 0.444 0.44
7 0.447 0.447 0.447 0.449 0.449 0.449 0.449 0.451 0.451 0.453 0.457 0.457 0.458
0.459 0.460 0.460 0.461 0.462 0.462 0.462 0.463 0.465 0.466 0.466 0.467 0.468 0.
473 0.473 0.473 0.474 0.475 0.476 0.476 0.479 0.488 0.488 0.492 0.492 0.492 0.49
3 0.494 0.494 0.496 0.496 0.498 0.499 0.501 0.504 0.506 0.507 0.507 0.510 0.511
0.512 0.518 0.518 0.521 0.523 0.523 0.523 0.523 0.524 0.529 0.530 0.530 0.531 0.
531 0.533 0.534 0.534 0.535 0.536 0.537 0.537 0.537 0.539 0.539 0.541 0.542 0.54
2 0.543 0.544 0.545 0.547 0.547 0.548 0.549 0.554 0.555 0.556 0.557 0.557 0.557
0.558 0.559 0.560 0.562 0.563 0.563 0.564 0.565 0.566 0.567 0.569 0.569 0.574 0.
575 0.576 0.577 0.578 0.582 0.582 0.584 0.588 0.588 0.589 0.590 0.594 0.594 0.59
5 0.597 0.597 0.599 0.600 0.603 0.604 0.604 0.606 0.614 0.616 0.616 0.619 0.620
0.622 0.622 0.624 0.625 0.626 0.629 0.633 0.634 0.637 0.638 0.640 0.641 0.643 0.
643 0.644 0.645 0.650 0.652 0.653 0.653 0.656 0.656 0.658 0.659 0.662 0.662 0.66
5 0.668 0.669 0.672 0.680 0.680 0.681 0.683 0.685 0.691 0.693 0.696 0.698 0.702
0.704 0.708 0.717 0.723 0.726 0.727 0.728 0.731 0.731 0.736 0.741 0.745 0.750 0.
765 0.768 0.771 0.772 0.774 0.781 0.784 0.784 0.803 0.806 0.809 0.813 0.814 0.81

9 0.824 0.828 0.838 0.840 0.846 0.846 0.856 0.858 0.867 0.872 0.884 0.884 0.889
0.889 0.901 0.915 0.917 0.947 0.950 0.955

Пример сгенерированной выборки длины 600: 0.030 0.040 0.045 0.048 0.050 0.058 0.
066 0.070 0.072 0.075 0.077 0.077 0.079 0.091 0.092 0.095 0.096 0.102 0.103 0.10
4 0.106 0.109 0.109 0.110 0.126 0.131 0.133 0.135 0.141 0.144 0.148 0.148 0.150
0.151 0.153 0.153 0.157 0.157 0.166 0.170 0.171 0.172 0.176 0.177 0.178 0.180 0.
183 0.185 0.188 0.192 0.194 0.197 0.198 0.199 0.199 0.201 0.202 0.202 0.203 0.20
3 0.209 0.211 0.214 0.214 0.216 0.216 0.219 0.220 0.221 0.222 0.223 0.227 0.228
0.228 0.232 0.233 0.235 0.235 0.236 0.237 0.237 0.238 0.238 0.238 0.238 0.239 0.
239 0.243 0.244 0.245 0.245 0.246 0.247 0.250 0.251 0.252 0.253 0.253 0.256 0.25
6 0.256 0.259 0.260 0.260 0.263 0.263 0.263 0.263 0.264 0.265 0.266 0.267 0.267
0.267 0.268 0.270 0.270 0.272 0.275 0.275 0.276 0.278 0.281 0.284 0.287 0.291 0.
292 0.295 0.298 0.298 0.299 0.300 0.301 0.302 0.302 0.303 0.303 0.304 0.308 0.30
9 0.309 0.310 0.311 0.314 0.315 0.317 0.319 0.320 0.323 0.324 0.325 0.326 0.328
0.330 0.330 0.330 0.331 0.331 0.332 0.332 0.333 0.334 0.334 0.334 0.335 0.337 0.
338 0.338 0.339 0.340 0.341 0.342 0.343 0.343 0.344 0.344 0.348 0.349 0.351 0.35
1 0.352 0.354 0.354 0.354 0.355 0.356 0.356 0.356 0.361 0.362 0.365 0.366 0.366
0.367 0.368 0.369 0.371 0.371 0.373 0.376 0.378 0.379 0.379 0.381 0.386 0.387 0.
391 0.392 0.392 0.392 0.393 0.394 0.396 0.396 0.399 0.399 0.399 0.400 0.401 0.40
2 0.405 0.405 0.405 0.405 0.406 0.407 0.407 0.409 0.409 0.410 0.412 0.412 0.413
0.414 0.414 0.415 0.415 0.417 0.420 0.421 0.421 0.422 0.423 0.423 0.425 0.425 0.
425 0.426 0.427 0.427 0.428 0.429 0.429 0.430 0.431 0.433 0.435 0.435 0.436 0.43
6 0.436 0.437 0.438 0.438 0.438 0.438 0.440 0.441 0.441 0.441 0.443 0.444 0.446
0.446 0.447 0.449 0.451 0.451 0.451 0.452 0.455 0.455 0.455 0.457 0.457 0.457 0.
457 0.459 0.459 0.459 0.459 0.459 0.460 0.460 0.460 0.461 0.461 0.461 0.461 0.46
1 0.463 0.463 0.463 0.464 0.464 0.464 0.465 0.466 0.467 0.470 0.470 0.471 0.471
0.473 0.473 0.475 0.475 0.477 0.478 0.482 0.483 0.483 0.484 0.485 0.485 0.487 0.
489 0.489 0.489 0.490 0.490 0.490 0.491 0.491 0.492 0.492 0.495 0.495 0.496 0.49
7 0.498 0.498 0.501 0.501 0.503 0.504 0.506 0.508 0.508 0.508 0.510 0.510 0.510
0.511 0.512 0.512 0.515 0.518 0.519 0.519 0.519 0.520 0.522 0.522 0.523 0.523 0.
523 0.523 0.524 0.526 0.530 0.532 0.533 0.534 0.534 0.534 0.534 0.534 0.536 0.53
6 0.537 0.537 0.538 0.538 0.540 0.541 0.545 0.546 0.547 0.548 0.548 0.550 0.551
0.553 0.553 0.554 0.556 0.556 0.557 0.558 0.558 0.562 0.565 0.565 0.568 0.571 0.
571 0.572 0.573 0.574 0.575 0.575 0.576 0.576 0.576 0.579 0.579 0.581 0.586 0.58
8 0.592 0.592 0.592 0.594 0.595 0.597 0.598 0.601 0.602 0.604 0.610 0.611 0.612
0.612 0.612 0.613 0.615 0.619 0.620 0.620 0.620 0.622 0.623 0.623 0.624 0.627 0.
627 0.628 0.630 0.630 0.630 0.630 0.633 0.635 0.637 0.637 0.639 0.641 0.642 0.64
3 0.643 0.644 0.646 0.646 0.647 0.647 0.647 0.648 0.649 0.651 0.654 0.654 0.655
0.656 0.656 0.657 0.661 0.661 0.666 0.668 0.669 0.670 0.673 0.675 0.678 0.679 0.
680 0.680 0.681 0.681 0.684 0.686 0.688 0.689 0.692 0.692 0.693 0.693 0.694 0.69
6 0.698 0.698 0.699 0.699 0.701 0.704 0.705 0.709 0.709 0.711 0.713 0.713 0.714
0.714 0.715 0.718 0.720 0.720 0.722 0.723 0.724 0.724 0.725 0.726 0.729 0.729 0.
735 0.737 0.740 0.744 0.745 0.745 0.745 0.747 0.750 0.752 0.755 0.755 0.755 0.75
8 0.759 0.760 0.764 0.767 0.767 0.767 0.768 0.771 0.774 0.777 0.778 0.786 0.787
0.788 0.788 0.790 0.792 0.797 0.800 0.810 0.811 0.816 0.818 0.820 0.823 0.825 0.
826 0.830 0.832 0.835 0.838 0.842 0.855 0.857 0.860 0.864 0.865 0.881 0.882 0.88
6 0.887 0.890 0.892 0.894 0.896 0.896 0.902 0.904 0.908 0.909 0.909 0.912 0.920
0.923 0.928 0.929 0.962 0.964 0.969 0.980

Пример сгенерированной выборки длины 800: 0.010 0.027 0.047 0.053 0.060 0.062 0.
068 0.069 0.070 0.079 0.079 0.084 0.088 0.093 0.097 0.098 0.103 0.105 0.111 0.11
1 0.113 0.113 0.115 0.115 0.122 0.124 0.125 0.126 0.133 0.133 0.134 0.135 0.138
0.139 0.139 0.140 0.140 0.140 0.140 0.140 0.141 0.150 0.153 0.154 0.157 0.159 0.
159 0.163 0.163 0.166 0.168 0.168 0.170 0.170 0.171 0.171 0.171 0.173 0.175 0.17
5 0.175 0.176 0.179 0.180 0.183 0.189 0.190 0.192 0.192 0.192 0.193 0.195 0.197
0.197 0.197 0.198 0.198 0.201 0.201 0.202 0.203 0.205 0.205 0.206 0.206 0.207 0.
209 0.210 0.212 0.212 0.213 0.214 0.220 0.221 0.222 0.223 0.225 0.225 0.225 0.22
5 0.227 0.229 0.230 0.232 0.233 0.233 0.234 0.235 0.235 0.236 0.236 0.237 0.239
0.239 0.239 0.240 0.242 0.243 0.247 0.248 0.251 0.253 0.253 0.255 0.259 0.262 0.
263 0.263 0.267 0.268 0.268 0.269 0.270 0.270 0.272 0.274 0.274 0.275 0.277 0.27
8 0.279 0.280 0.281 0.282 0.283 0.284 0.285 0.285 0.285 0.286 0.286 0.286 0.288

0.290 0.291 0.291 0.292 0.292 0.293 0.294 0.294 0.294 0.294 0.294 0.295 0.295 0.
299 0.300 0.300 0.301 0.304 0.305 0.308 0.308 0.308 0.308 0.309 0.310 0.312 0.31
3 0.314 0.315 0.316 0.317 0.318 0.319 0.320 0.320 0.321 0.323 0.324 0.325 0.327
0.328 0.333 0.333 0.333 0.336 0.336 0.337 0.337 0.339 0.339 0.339 0.341 0.344 0.
345 0.346 0.347 0.347 0.348 0.348 0.349 0.350 0.351 0.352 0.354 0.354 0.355 0.35
6 0.362 0.363 0.363 0.365 0.365 0.365 0.368 0.370 0.371 0.371 0.373 0.374 0.376
0.377 0.377 0.377 0.378 0.378 0.380 0.382 0.382 0.382 0.382 0.384 0.384 0.384 0.
385 0.385 0.385 0.386 0.386 0.386 0.387 0.388 0.388 0.389 0.389 0.389 0.389 0.39
1 0.391 0.391 0.391 0.392 0.392 0.392 0.393 0.393 0.393 0.394 0.395 0.396 0.396
0.397 0.397 0.397 0.398 0.398 0.399 0.399 0.399 0.399 0.400 0.400 0.400 0.401 0.
401 0.404 0.404 0.405 0.405 0.405 0.406 0.406 0.408 0.409 0.411 0.413 0.413 0.41
3 0.414 0.414 0.414 0.415 0.415 0.416 0.417 0.417 0.417 0.418 0.418 0.419 0.420
0.421 0.426 0.427 0.427 0.427 0.428 0.428 0.428 0.428 0.430 0.431 0.432 0.432 0.
432 0.433 0.433 0.434 0.435 0.435 0.436 0.436 0.437 0.439 0.440 0.440 0.441 0.44
1 0.441 0.441 0.444 0.444 0.444 0.444 0.446 0.447 0.447 0.447 0.447 0.448 0.449
0.449 0.450 0.450 0.451 0.451 0.452 0.452 0.453 0.453 0.454 0.454 0.455 0.456 0.
457 0.457 0.458 0.458 0.459 0.459 0.459 0.459 0.460 0.462 0.462 0.462 0.462 0.46
4 0.464 0.465 0.465 0.465 0.465 0.467 0.467 0.467 0.468 0.468 0.468 0.469 0.470
0.470 0.470 0.470 0.470 0.471 0.472 0.473 0.473 0.474 0.474 0.475 0.475 0.476 0.
479 0.480 0.480 0.480 0.481 0.481 0.482 0.482 0.482 0.483 0.483 0.483 0.485 0.48
6 0.489 0.489 0.489 0.490 0.493 0.493 0.494 0.495 0.496 0.496 0.496 0.498 0.498
0.498 0.500 0.500 0.500 0.502 0.502 0.502 0.503 0.503 0.504 0.505 0.505 0.506 0.
506 0.507 0.507 0.508 0.508 0.508 0.508 0.508 0.508 0.509 0.510 0.511 0.511 0.51
3 0.513 0.513 0.513 0.514 0.517 0.517 0.518 0.518 0.521 0.522 0.522 0.522 0.522
0.523 0.523 0.524 0.524 0.524 0.525 0.525 0.526 0.528 0.528 0.530 0.530 0.531 0.
533 0.533 0.534 0.534 0.534 0.535 0.535 0.535 0.537 0.538 0.540 0.540 0.541 0.54
1 0.542 0.542 0.544 0.545 0.548 0.548 0.548 0.549 0.551 0.551 0.554 0.556 0.556
0.556 0.556 0.557 0.557 0.557 0.558 0.559 0.560 0.560 0.560 0.564 0.568 0.568 0.
568 0.568 0.569 0.569 0.569 0.570 0.570 0.570 0.570 0.571 0.574 0.575 0.578 0.57
9 0.580 0.581 0.582 0.583 0.583 0.584 0.584 0.584 0.585 0.587 0.587 0.587 0.589
0.589 0.590 0.590 0.591 0.592 0.592 0.593 0.593 0.594 0.594 0.594 0.595 0.595 0.
595 0.595 0.595 0.595 0.597 0.597 0.599 0.600 0.602 0.605 0.609 0.609 0.609 0.609 0.60
9 0.610 0.611 0.611 0.611 0.612 0.615 0.615 0.616 0.618 0.618 0.619 0.621 0.621
0.621 0.625 0.625 0.626 0.627 0.628 0.628 0.629 0.629 0.631 0.632 0.633 0.633 0.
633 0.634 0.636 0.636 0.637 0.637 0.637 0.639 0.639 0.641 0.642 0.642 0.642 0.64
3 0.643 0.644 0.645 0.645 0.651 0.652 0.654 0.656 0.657 0.658 0.659 0.659 0.660
0.660 0.661 0.662 0.662 0.663 0.663 0.663 0.663 0.664 0.664 0.667 0.667 0.668 0.
669 0.671 0.672 0.674 0.676 0.677 0.683 0.684 0.684 0.684 0.686 0.686 0.688 0.68
9 0.691 0.691 0.691 0.692 0.692 0.694 0.694 0.695 0.697 0.697 0.698 0.700 0.700
0.700 0.703 0.703 0.704 0.706 0.706 0.708 0.710 0.712 0.716 0.716 0.718 0.724 0.
724 0.726 0.726 0.727 0.727 0.728 0.728 0.729 0.733 0.733 0.734 0.736 0.736 0.73
9 0.740 0.741 0.741 0.742 0.742 0.745 0.745 0.745 0.748 0.749 0.749 0.749 0.750
0.751 0.752 0.753 0.753 0.756 0.760 0.762 0.762 0.765 0.766 0.767 0.769 0.769 0.
771 0.772 0.774 0.776 0.776 0.776 0.777 0.777 0.779 0.783 0.784 0.785 0.785 0.78
6 0.789 0.791 0.797 0.798 0.804 0.807 0.812 0.812 0.812 0.813 0.816 0.818 0.819
0.820 0.822 0.823 0.823 0.824 0.830 0.831 0.831 0.832 0.835 0.835 0.837 0.837 0.
839 0.846 0.853 0.855 0.859 0.859 0.860 0.862 0.862 0.864 0.876 0.879 0.882 0.88
9 0.892 0.896 0.897 0.905 0.905 0.908 0.911 0.912 0.914 0.916 0.918 0.918 0.918
0.922 0.932 0.936 0.941 0.946 0.962 0.977

Пример сгенерированной выборки длины 1000: 0.012 0.026 0.035 0.039 0.046 0.049
0.051 0.052 0.058 0.058 0.069 0.080 0.087 0.093 0.095 0.095 0.098 0.099 0.104 0.
105 0.105 0.106 0.113 0.114 0.115 0.116 0.119 0.119 0.120 0.122 0.129 0.129 0.12
9 0.130 0.131 0.132 0.133 0.136 0.136 0.138 0.139 0.141 0.141 0.142 0.145 0.146
0.148 0.148 0.149 0.152 0.153 0.155 0.158 0.160 0.164 0.167 0.168 0.168 0.169 0.
170 0.170 0.170 0.172 0.174 0.174 0.175 0.176 0.177 0.179 0.180 0.180 0.182 0.18
4 0.184 0.185 0.188 0.189 0.193 0.193 0.198 0.199 0.199 0.200 0.201 0.201 0.202
0.203 0.206 0.207 0.212 0.213 0.214 0.215 0.216 0.216 0.216 0.217 0.217 0.220 0.
221 0.221 0.222 0.222 0.223 0.226 0.229 0.230 0.230 0.231 0.231 0.231 0.232 0.23
2 0.232 0.235 0.236 0.237 0.239 0.240 0.240 0.242 0.242 0.242 0.242 0.243 0.244
0.246 0.247 0.248 0.249 0.250 0.250 0.250 0.252 0.254 0.254 0.255 0.255 0.256 0.
257 0.257 0.258 0.259 0.260 0.260 0.261 0.262 0.262 0.264 0.265 0.265 0.265 0.26

7 0.268 0.268 0.268 0.270 0.270 0.270 0.270 0.271 0.272 0.273 0.273 0.275 0.276
0.277 0.277 0.279 0.282 0.282 0.283 0.283 0.284 0.284 0.284 0.285 0.285 0.286 0.
286 0.286 0.287 0.288 0.288 0.290 0.292 0.293 0.293 0.293 0.293 0.293 0.295 0.29
7 0.297 0.298 0.301 0.302 0.302 0.303 0.304 0.304 0.306 0.308 0.308 0.310 0.310
0.310 0.311 0.313 0.313 0.314 0.317 0.318 0.318 0.321 0.321 0.322 0.322 0.322 0.
323 0.323 0.324 0.324 0.324 0.324 0.324 0.325 0.326 0.326 0.327 0.327 0.327 0.32
7 0.327 0.327 0.329 0.331 0.331 0.331 0.332 0.332 0.333 0.333 0.333 0.334 0.334
0.336 0.336 0.336 0.337 0.337 0.339 0.340 0.340 0.340 0.341 0.342 0.342 0.342 0.
342 0.342 0.343 0.343 0.343 0.344 0.344 0.345 0.345 0.346 0.346 0.346 0.346 0.34
6 0.347 0.347 0.347 0.348 0.348 0.350 0.350 0.353 0.353 0.353 0.353 0.353 0.354
0.355 0.357 0.360 0.360 0.361 0.362 0.362 0.362 0.362 0.364 0.364 0.364 0.366 0.
366 0.366 0.367 0.367 0.368 0.369 0.370 0.370 0.373 0.373 0.374 0.374 0.374 0.37
4 0.374 0.375 0.376 0.377 0.377 0.378 0.380 0.380 0.381 0.382 0.383 0.383 0.383
0.383 0.384 0.384 0.385 0.385 0.386 0.387 0.388 0.389 0.389 0.390 0.390 0.391 0.
392 0.393 0.393 0.393 0.393 0.395 0.395 0.395 0.396 0.397 0.397 0.397 0.399 0.39
9 0.399 0.401 0.401 0.401 0.401 0.402 0.403 0.403 0.404 0.405 0.406 0.407 0.407
0.408 0.409 0.410 0.410 0.411 0.411 0.411 0.411 0.412 0.412 0.412 0.412 0.413 0.
413 0.414 0.414 0.414 0.415 0.415 0.415 0.416 0.416 0.417 0.419 0.419 0.419 0.42
0 0.420 0.420 0.421 0.421 0.422 0.422 0.423 0.424 0.425 0.427 0.427 0.428 0.428
0.428 0.428 0.428 0.429 0.430 0.430 0.430 0.430 0.433 0.433 0.433 0.433 0.434 0.
435 0.438 0.438 0.438 0.438 0.439 0.440 0.440 0.440 0.441 0.442 0.442 0.442 0.44
2 0.443 0.443 0.445 0.445 0.445 0.445 0.445 0.445 0.446 0.446 0.446 0.446 0.447
0.447 0.448 0.448 0.448 0.448 0.449 0.451 0.452 0.453 0.453 0.454 0.455 0.455 0.
456 0.456 0.456 0.456 0.457 0.459 0.460 0.461 0.461 0.462 0.462 0.462 0.462 0.46
4 0.464 0.464 0.464 0.464 0.465 0.465 0.466 0.466 0.466 0.466 0.467 0.467 0.468
0.468 0.468 0.470 0.470 0.470 0.471 0.472 0.472 0.472 0.472 0.472 0.473 0.474 0.
474 0.474 0.475 0.475 0.475 0.476 0.476 0.477 0.477 0.478 0.478 0.478 0.481 0.48
2 0.484 0.484 0.487 0.487 0.487 0.488 0.489 0.489 0.489 0.490 0.490 0.490 0.491
0.491 0.492 0.492 0.493 0.495 0.496 0.496 0.497 0.497 0.497 0.498 0.498 0.498 0.
499 0.499 0.500 0.500 0.501 0.501 0.502 0.502 0.503 0.503 0.503 0.504 0.504 0.50
6 0.506 0.506 0.507 0.507 0.508 0.509 0.509 0.509 0.510 0.510 0.511 0.513 0.514
0.516 0.516 0.516 0.517 0.518 0.519 0.520 0.521 0.523 0.523 0.524 0.524 0.525 0.
525 0.526 0.527 0.528 0.529 0.529 0.529 0.530 0.530 0.532 0.532 0.532 0.533 0.53
3 0.534 0.535 0.535 0.536 0.536 0.538 0.539 0.539 0.539 0.540 0.540 0.540 0.541
0.541 0.541 0.542 0.543 0.543 0.543 0.545 0.545 0.546 0.546 0.547 0.547 0.548 0.
548 0.548 0.549 0.549 0.551 0.551 0.551 0.552 0.553 0.553 0.554 0.554 0.555 0.55
6 0.556 0.557 0.557 0.557 0.557 0.558 0.559 0.559 0.559 0.560 0.560 0.560 0.560
0.560 0.561 0.561 0.562 0.562 0.562 0.562 0.563 0.564 0.564 0.566 0.566 0.567 0.
567 0.568 0.569 0.569 0.570 0.570 0.570 0.571 0.571 0.571 0.571 0.571 0.573 0.57
4 0.575 0.576 0.576 0.576 0.578 0.578 0.579 0.580 0.580 0.583 0.584 0.585 0.586
0.588 0.589 0.589 0.591 0.592 0.593 0.594 0.595 0.596 0.596 0.597 0.598 0.598 0.
599 0.600 0.601 0.601 0.604 0.605 0.606 0.606 0.607 0.607 0.612 0.613 0.613 0.61
3 0.614 0.617 0.617 0.618 0.619 0.620 0.621 0.621 0.623 0.623 0.624 0.625 0.625
0.626 0.626 0.628 0.628 0.628 0.629 0.632 0.633 0.633 0.634 0.634 0.634 0.635 0.
635 0.635 0.635 0.636 0.636 0.638 0.638 0.639 0.639 0.641 0.642 0.643 0.643 0.64
4 0.644 0.645 0.646 0.647 0.647 0.647 0.647 0.648 0.649 0.649 0.652 0.653 0.653
0.654 0.657 0.658 0.658 0.659 0.659 0.660 0.661 0.663 0.663 0.664 0.664 0.664 0.
666 0.667 0.667 0.667 0.669 0.669 0.669 0.671 0.672 0.673 0.675 0.675 0.675 0.67
6 0.676 0.677 0.677 0.677 0.678 0.680 0.681 0.682 0.682 0.683 0.684 0.684 0.685
0.685 0.686 0.688 0.689 0.690 0.692 0.694 0.695 0.696 0.697 0.699 0.700 0.700 0.
700 0.701 0.703 0.704 0.704 0.705 0.706 0.708 0.709 0.709 0.710 0.710 0.710 0.71
0 0.711 0.712 0.712 0.713 0.713 0.715 0.715 0.715 0.716 0.716 0.717 0.717 0.718
0.719 0.720 0.721 0.722 0.723 0.723 0.725 0.727 0.727 0.728 0.728 0.730 0.731 0.
731 0.733 0.733 0.734 0.735 0.735 0.736 0.736 0.738 0.738 0.739 0.739 0.739 0.74
0 0.741 0.743 0.744 0.745 0.745 0.747 0.747 0.749 0.749 0.753 0.753 0.753 0.755
0.757 0.761 0.765 0.765 0.767 0.767 0.767 0.770 0.770 0.771 0.771 0.771 0.774 0.
774 0.777 0.777 0.778 0.780 0.782 0.786 0.787 0.788 0.790 0.790 0.790 0.790 0.79
0 0.791 0.793 0.795 0.795 0.795 0.797 0.798 0.799 0.802 0.803 0.806 0.807 0.810
0.810 0.810 0.810 0.810 0.813 0.813 0.813 0.814 0.818 0.818 0.820 0.820 0.822 0.
824 0.826 0.826 0.827 0.828 0.828 0.834 0.834 0.835 0.836 0.837 0.838 0.840 0.84
1 0.841 0.847 0.848 0.848 0.848 0.852 0.854 0.854 0.855 0.855 0.856 0.859 0.863
0.868 0.869 0.871 0.873 0.873 0.873 0.876 0.880 0.880 0.888 0.890 0.891 0.894 0.

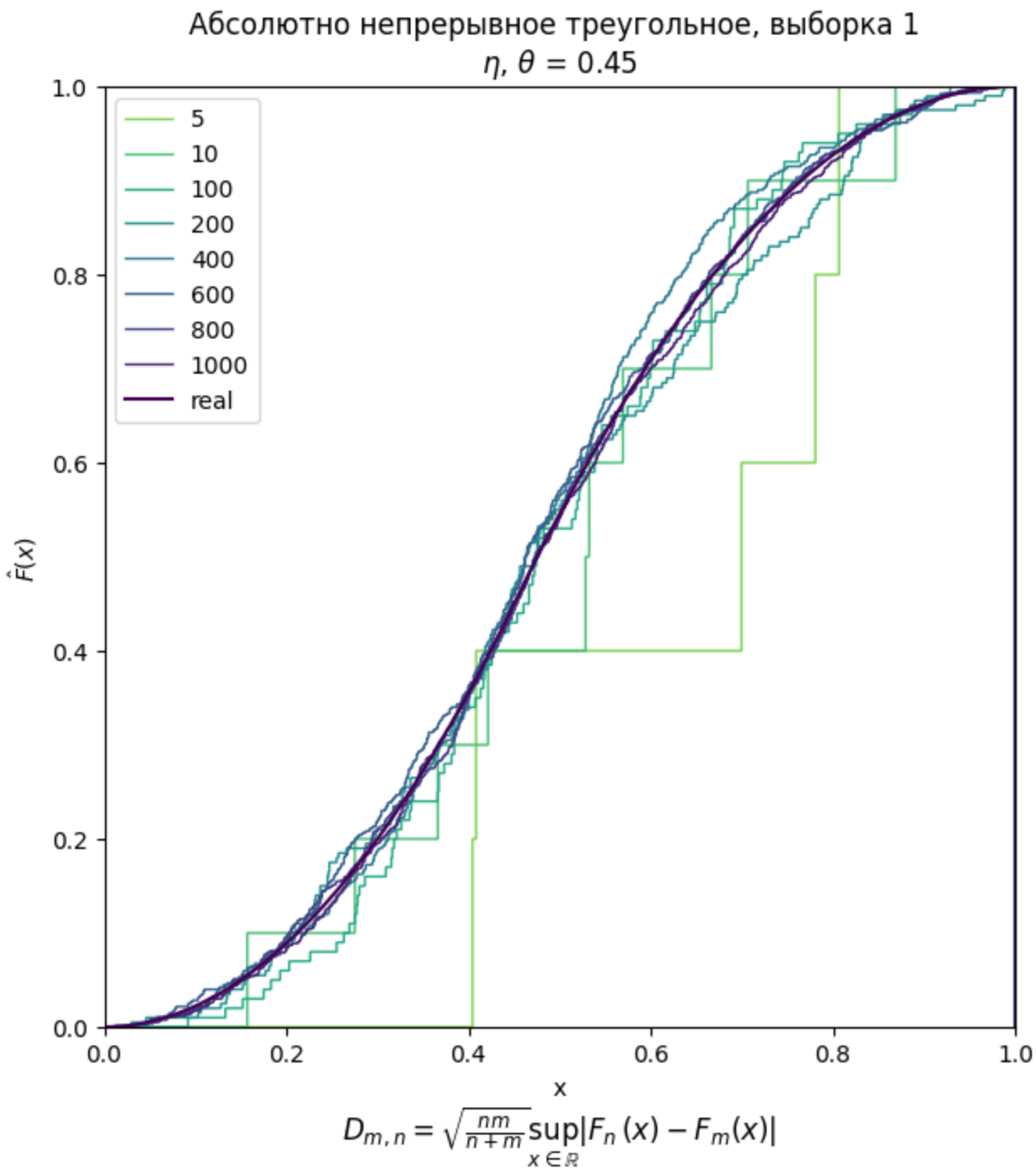
```
897 0.899 0.900 0.901 0.904 0.912 0.912 0.913 0.914 0.919 0.920 0.925 0.925 0.93
7 0.937 0.938 0.947 0.953 0.955 0.969 0.983
----------
```

Задание 2

In [24]:

```
def eta_distr(sample, x):
    res = 0
    for i in sample:
        if i <= x:
            res += 1
    return res / len(sample)

def eta_distr_real(x, theta=(0.45)):
    if x<0: return 0
    if x<theta: return x*x/theta
    if x<1: return (2*x-x*x-theta)/(1-theta)
    return 1

X_realeta = np.linspace(0,1,1000)
Y_realeta = np.array([eta_distr_real(x) for x in X_realeta])

Yeta = np.array([[[eta_distr(sample_eta[k][j], x) for x in X_realeta]
                  for j in range(5)] for k in range(len(n))])

def eta_Dmn(Yn, Ym, n, m):
    res = 0
    for i in range(29):
        d = abs(Yn[i]-Ym[i])
        if d>res: res = d
    return (n*m/(n+m))**0.5*res

diffseta = np.array([[[('%.2f' % eta_Dmn(Yeta[i][k], Yeta[j][k], n[i], n[i]))
                       if i>j else '-' for i in range(len(n))]
                      for j in range(len(n))]
                     for k in range(5)])
```

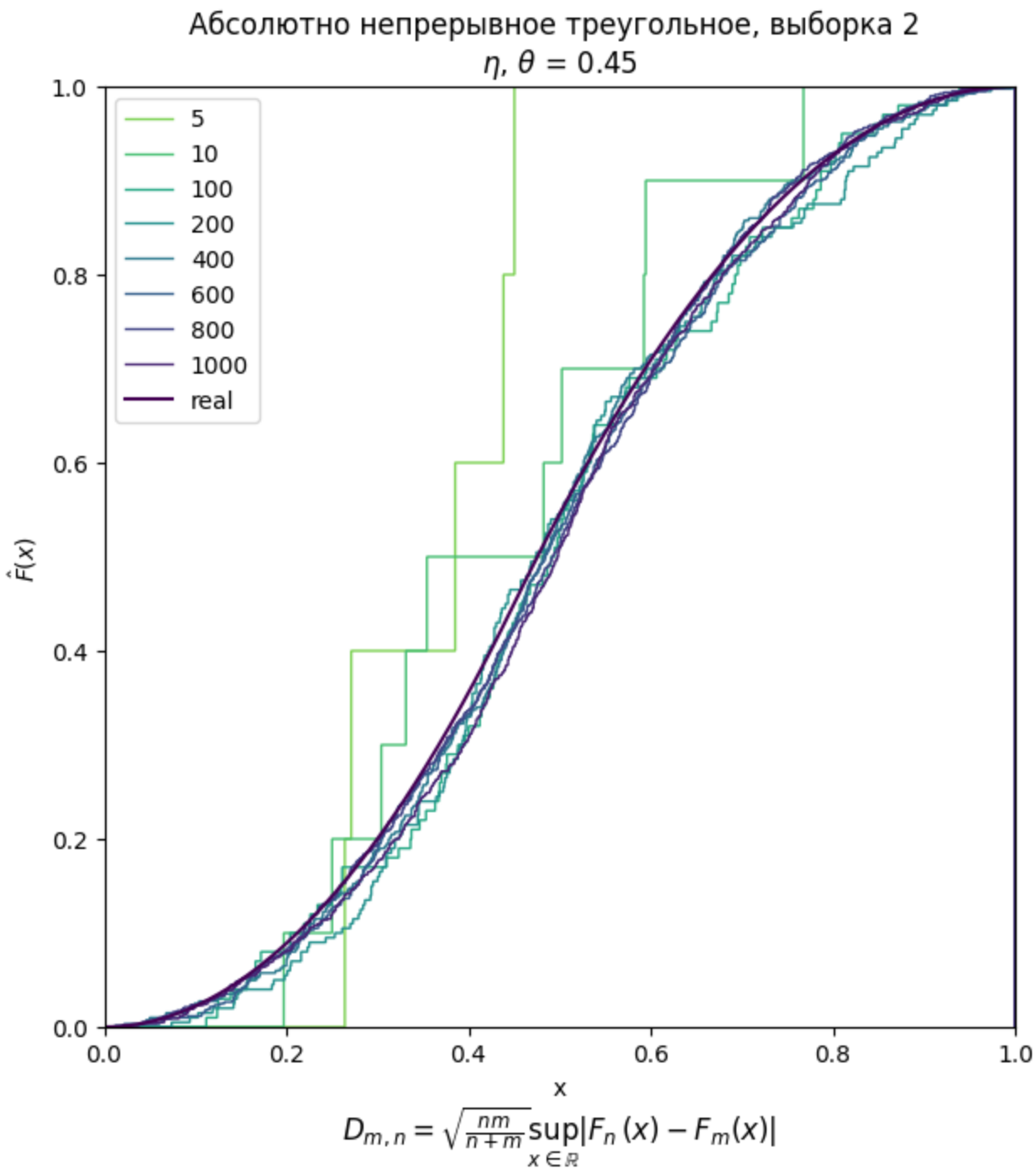
In [25]:
```
colors = ['#7bd152', '#45be71', '#25a885', '#21908c',
          '#2b798e', '#355f8d', '#414486', '#482574']

# graph[0]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yeta[i][0], np.append(X_realeta, 1), color = colors[i])
ax[0].plot(X_realeta, Y_realeta, color = '#440154')
ax[0].set(xmargin = 0, ymargin = 0, xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Абсолютно непрерывное треугольное, выборка 1 \n' +
                  '$\\eta , \\,\\theta$ = 0.45')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[0], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$' +
                r'$|F_n(x)-F_m(x)|$');
```



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	2.76	3.60	5.87	6.78	7.97	8.56
10	-	-	0.92	1.05	1.63	1.76	1.98	2.50
100	-	-	-	1.00	0.88	0.87	1.37	1.16
200	-	-	-	-	0.92	1.13	1.47	1.48
400	-	-	-	-	-	0.49	0.45	1.22
600	-	-	-	-	-	-	0.61	0.98
800	-	-	-	-	-	-	-	0.92
1000	-	-	-	-	-	-	-	-

In [26]:
```
# graph[1]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yeta[i][1], np.append(X_realeta, 1), color = colors[i])
ax[0].plot(X_realeta, Y_realeta, color = '#440154')
ax[0].set(xmargin = 0, ymargin = 0, xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Абсолютно непрерывное треугольное, выборка 2 \n' +
                  '$\\eta , \\,\\theta$ = 0.45')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[1], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$' +
                r'$|F_n(x)-F_m(x)|$');
```

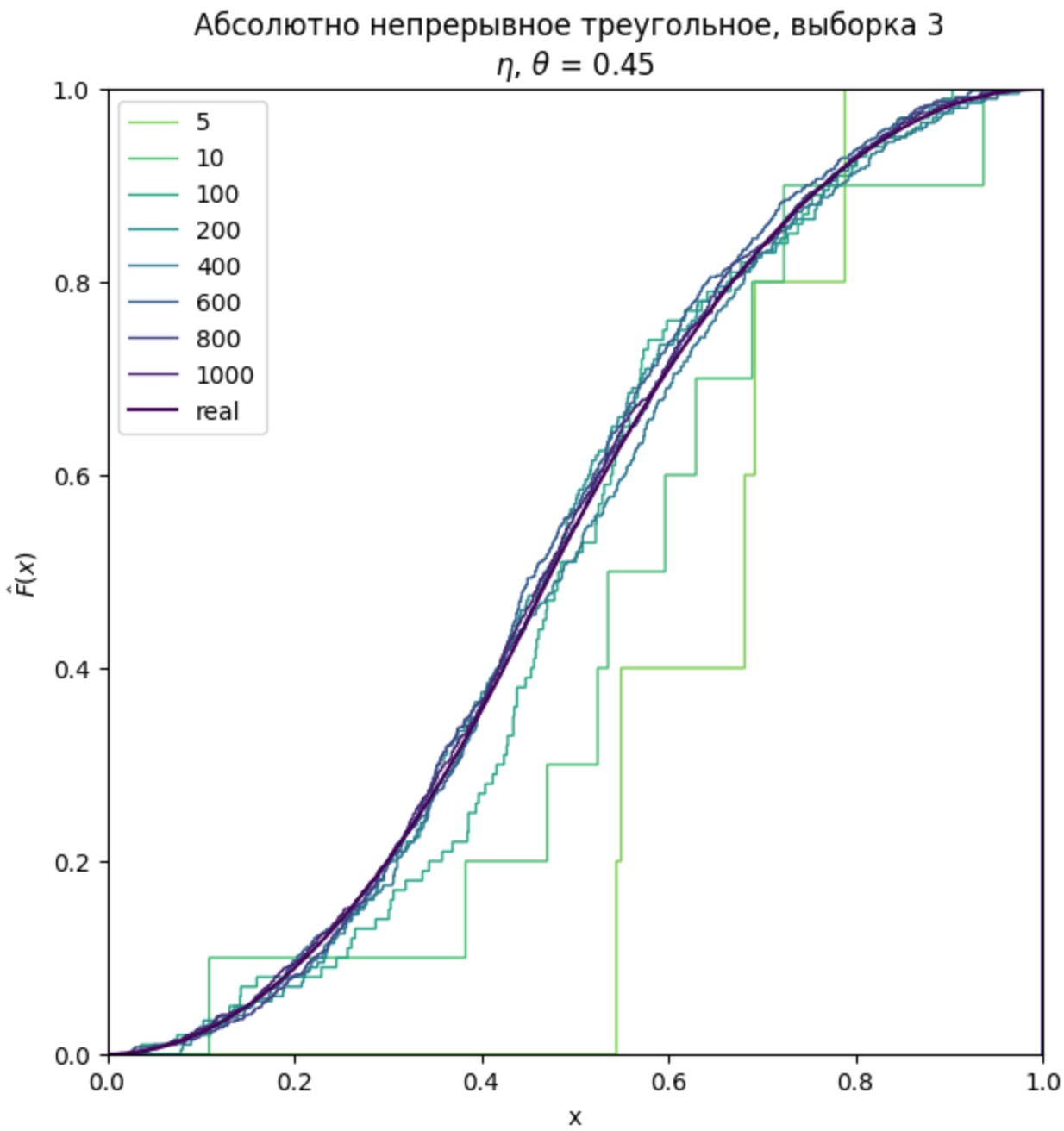


$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	1.12	1.63	2.30	3.18	4.30	4.65	5.28
10	-	-	2.62	3.50	4.24	6.03	6.60	7.69
100	-	-	-	0.75	1.38	1.15	1.65	1.45
200	-	-	-	-	1.27	0.61	1.22	0.56
400	-	-	-	-	-	1.37	0.95	1.78
600	-	-	-	-	-	-	0.82	0.48
800	-	-	-	-	-	-	-	0.85
1000	-	-	-	-	-	-	-	-

In [27]:

```
# graph[2]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yeta[i][2], np.append(X_realeta, 1), color = colors[i])
ax[0].plot(X_realeta, Y_realeta, color = '#440154')
ax[0].set(xmargin = 0, ymargin = 0, xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Абсолютно непрерывное треугольное, выборка 3 \n' +
                  '$\\eta , \\,\\theta$ = 0.45')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[2], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$' +
                r'$|F_n(x)-F_m(x)|$');
```

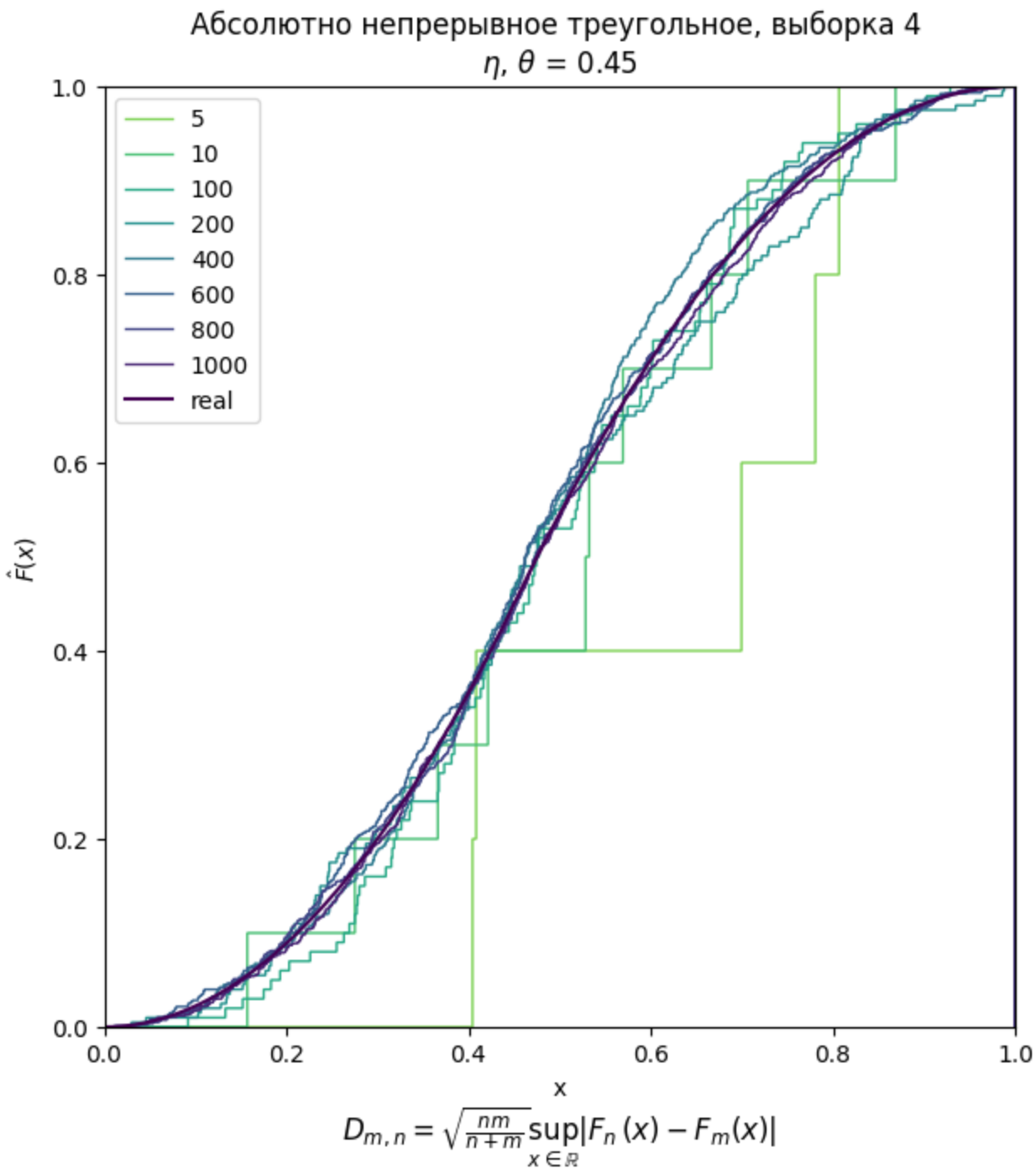


$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	1.41	1.85	2.69	2.68	3.38	3.53
10	-	-	1.77	2.60	3.32	4.79	5.05	6.31
100	-	-	-	1.00	1.10	1.33	1.20	1.88
200	-	-	-	-	1.03	0.78	1.13	0.92
400	-	-	-	-	-	0.75	0.62	1.13
600	-	-	-	-	-	-	0.79	0.70
800	-	-	-	-	-	-	-	0.99
1000	-	-	-	-	-	-	-	-

In [28]:

```
# graph[3]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yeta[i][0], np.append(X_realeta, 1), color = colors[i])
ax[0].plot(X_realeta, Y_realeta, color = '#440154')
ax[0].set(xmargin = 0, ymargin = 0, xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Абсолютно непрерывное треугольное, выборка 4 \n' +
                  '$\\eta , \\,\\theta$ = 0.45')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[3], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$' +
                r'$|F_n(x)-F_m(x)|$');
```

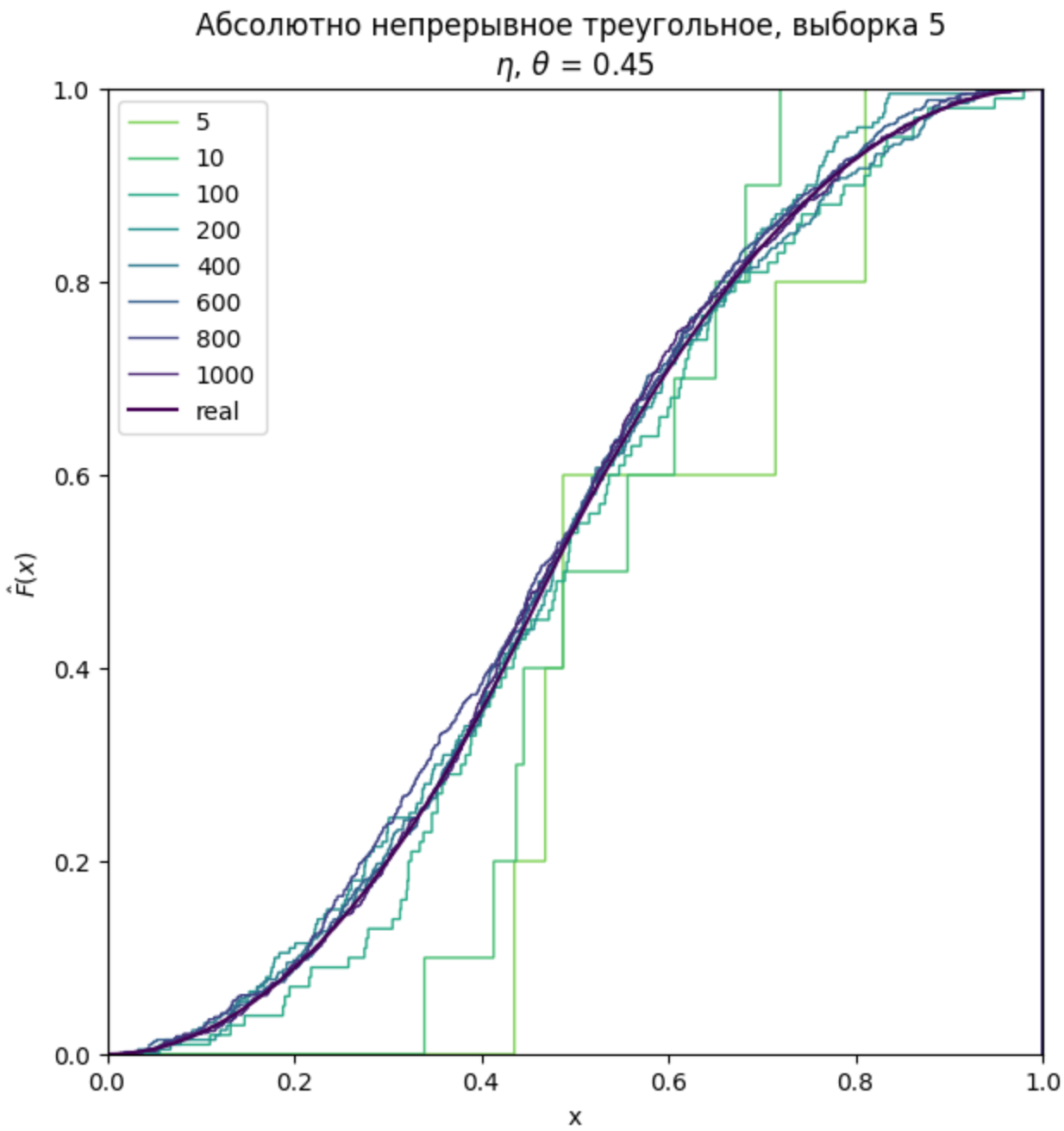


$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	2.69	3.55	5.06	6.12	7.33	8.05
10	-	-	0.99	1.30	1.94	2.25	2.65	2.84
100	-	-	-	0.85	1.52	1.50	1.77	2.15
200	-	-	-	-	0.74	0.69	1.15	1.12
400	-	-	-	-	-	0.46	0.57	0.44
600	-	-	-	-	-	-	0.48	0.45
800	-	-	-	-	-	-	-	0.44
1000	-	-	-	-	-	-	-	-

In [29]:

```
# graph[4]
fig, ax = plt.subplots(2,1, figsize=(7,12), height_ratios = [2,1])
for i in range(8):
    ax[0].stairs(Yeta[i][4], np.append(X_realeta, 1), color = colors[i])
ax[0].plot(X_realeta, Y_realeta, color = '#440154')
ax[0].set(xmargin = 0, ymargin = 0, xlabel = 'x', ylabel = r'$\hat{F}(x)$',
          title = 'Абсолютно непрерывное треугольное, выборка 5 \n' +
                  '$\\eta , \\,\\theta$ = 0.45')
ax[0].legend([*n, 'real'], loc='upper left');

ax[1].table(cellText = diffsxi[4], rowLabels=n, colLabels=n,
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title(r'$D_{m,n}=\sqrt{\frac{nm}{n+m}}\sup_{x\in\mathbb{R}}$' +
                r'$|F_n(x)-F_m(x)|$');
```



$$D_{m,n} = \sqrt{\frac{nm}{n+m}} \sup_{x \in \mathbb{R}} |F_n(x) - F_m(x)|$$

	5	10	100	200	400	600	800	1000
5	-	0.67	1.27	2.50	3.18	4.21	4.68	4.99
10	-	-	0.92	1.60	2.02	2.60	3.15	3.20
100	-	-	-	1.15	0.95	1.10	1.55	1.41
200	-	-	-	-	1.56	1.41	1.98	1.57
400	-	-	-	-	-	0.92	0.85	0.91
600	-	-	-	-	-	-	1.45	0.57
800	-	-	-	-	-	-	-	1.40
1000	-	-	-	-	-	-	-	-

Задание 3

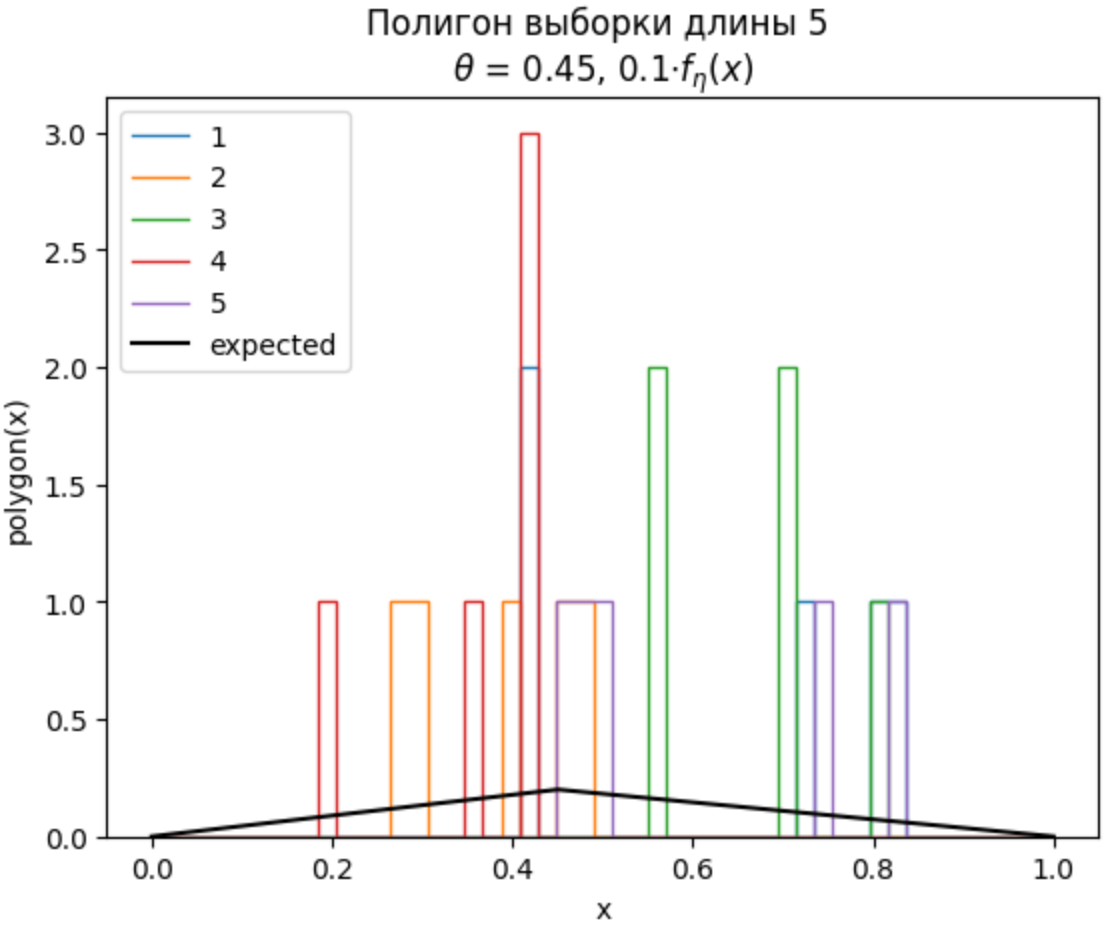
Логика с домножением вероятности здесь также применима, однако она должна быть немного модифицирована: полигон частот как таковой почти не будет иметь смысла, так как вероятность попадания в каждое значение стремится к нулю, что значит что если просто расставить на отрезке от 0 до 1 все значения выборки, то очень наврядли полученный график поднимется выше единицы. Получить хоть какой-то смысл из этого графика будет затруднительно, потому нужно из него будет сложно получить что-то больше чем прямую, не то что сравнить с потенциально сложной функцией вероятности.

Для решения этой проблемы имеет смысл разбить данный отрезок на сколько-то частей. Да, это примерно сведет случай к дискретному, однако это даст весьма уверенную возможность проанализировать график полигона частот относительно графика вероятности. И вот где логика этого номера с дискретной вероятностью ломается: нам не известна функциц я вероятности, нам дана плотность распределения. Но и эта проблема теперь решается достаточно просто: если рассматривать не саму плотность вероятности, а плотность вероятности на каком-то небольшом участке, то тогда плотность вероятности домноженная на длину рассматриваемого участка уже должна соотноситься с формой эмперической вероятности, определенной ранее как $\hat{P}(x)$:

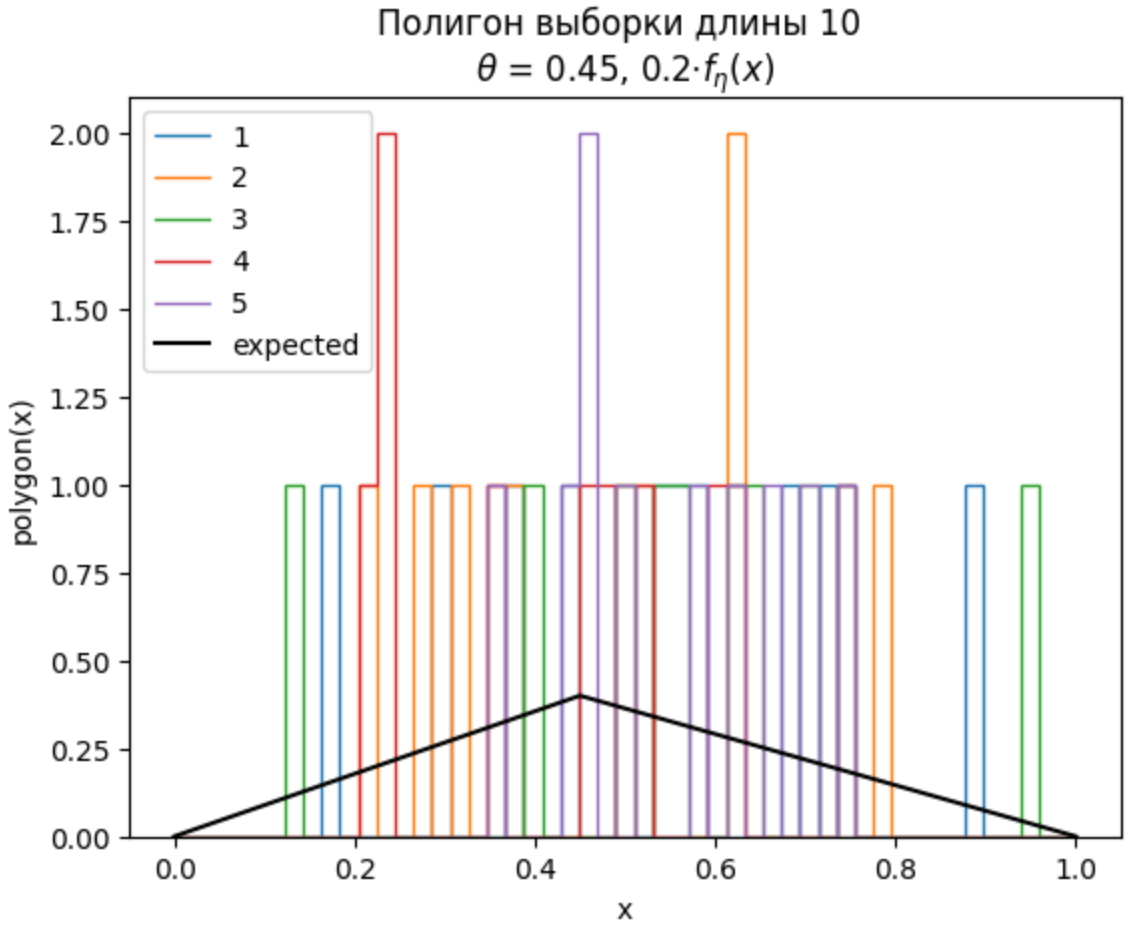
При $n \to \infty \, \Delta x f(x) = \frac{k}{n} \Rightarrow k = n \Delta x f(x)$. В этом задании я разделил отрезок от 0 до 1 на 50 равных между собой отрезков. Тогда полигон частот должен соотноситься с плотностью через коэффициент домножения плотности $\frac{n = \text{длина выборки}}{50}$

In [30]:
```
def eta_pilygon(sample, X):
    Y = np.zeros(X.shape)
    tick = 1
    for i in sample:
        while i > X[tick]: tick+=1
        Y[tick]+=1
    return Y

def eta_posib(x, theta = 0.45):
    if x<0: return 0
    if x<theta: return 2*x/theta
    if x<1: return 2*(1-x)/(1-theta)
    return 0

X_poleta = np.linspace(0,1,50)
Y_poleta = [[eta_pilygon(sample_eta[k][j], X_poleta) for j in range(5)]
            for k in range(len(n))]

posibilityeta = np.array([eta_posib(x) for x in X_realeta])
```

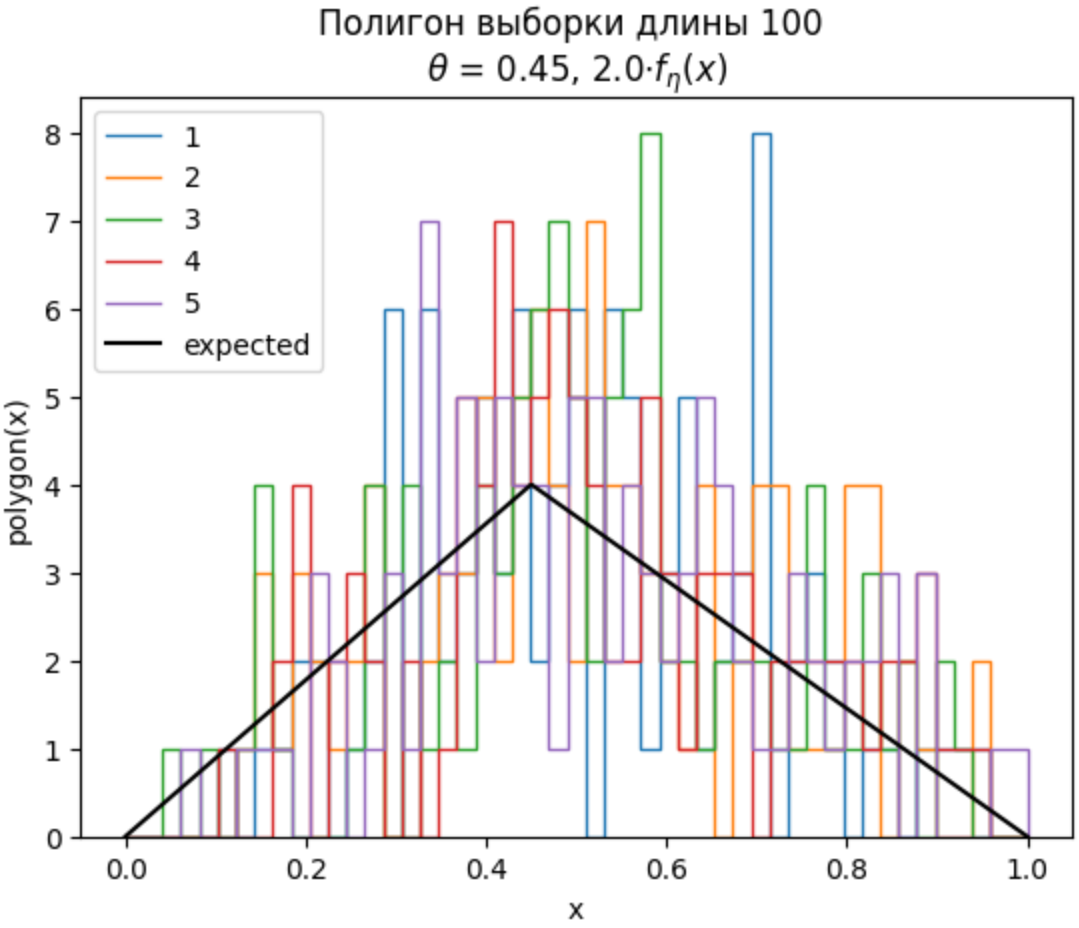
In [31]:
```
# n=5
for i in range(5):
    plt.stairs(Y_poleta[0][i], np.append(X_poleta,1))
plt.plot(X_realeta, posibilityeta*5/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 5 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(5/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



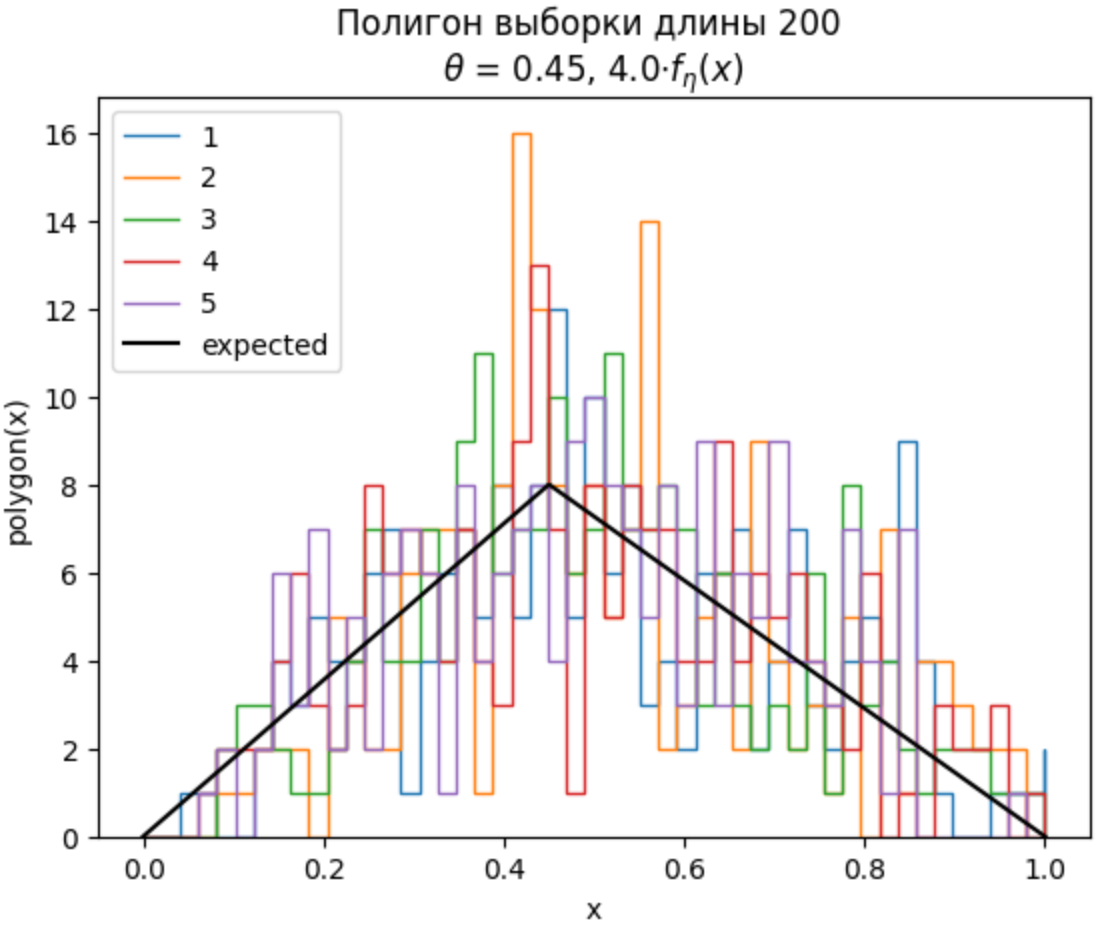
In [32]:
```
# n=10
for i in range(5):
    plt.stairs(Y_poleta[1][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*10/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 10 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(10/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



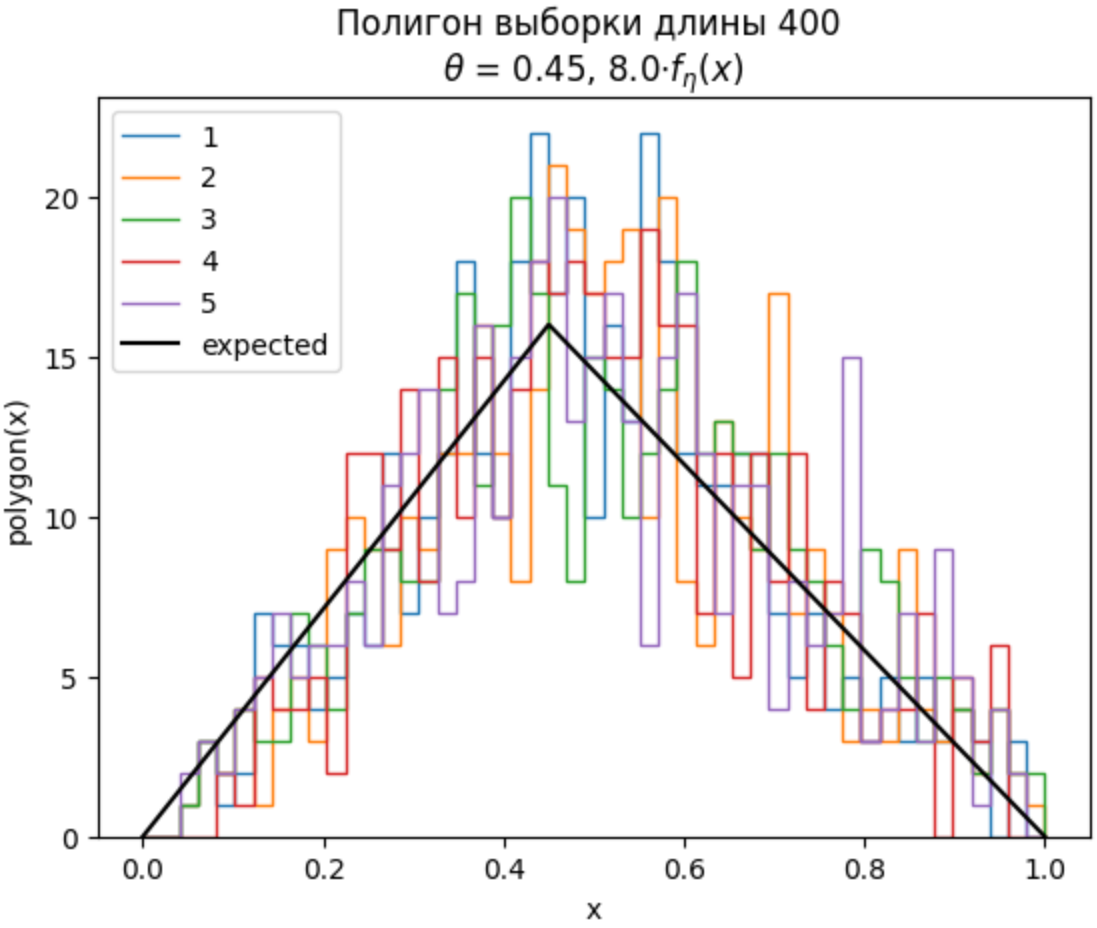
In [33]:
```
# n=100
for i in range(5):
    plt.stairs(Y_poleta[2][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*100/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 100 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(100/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



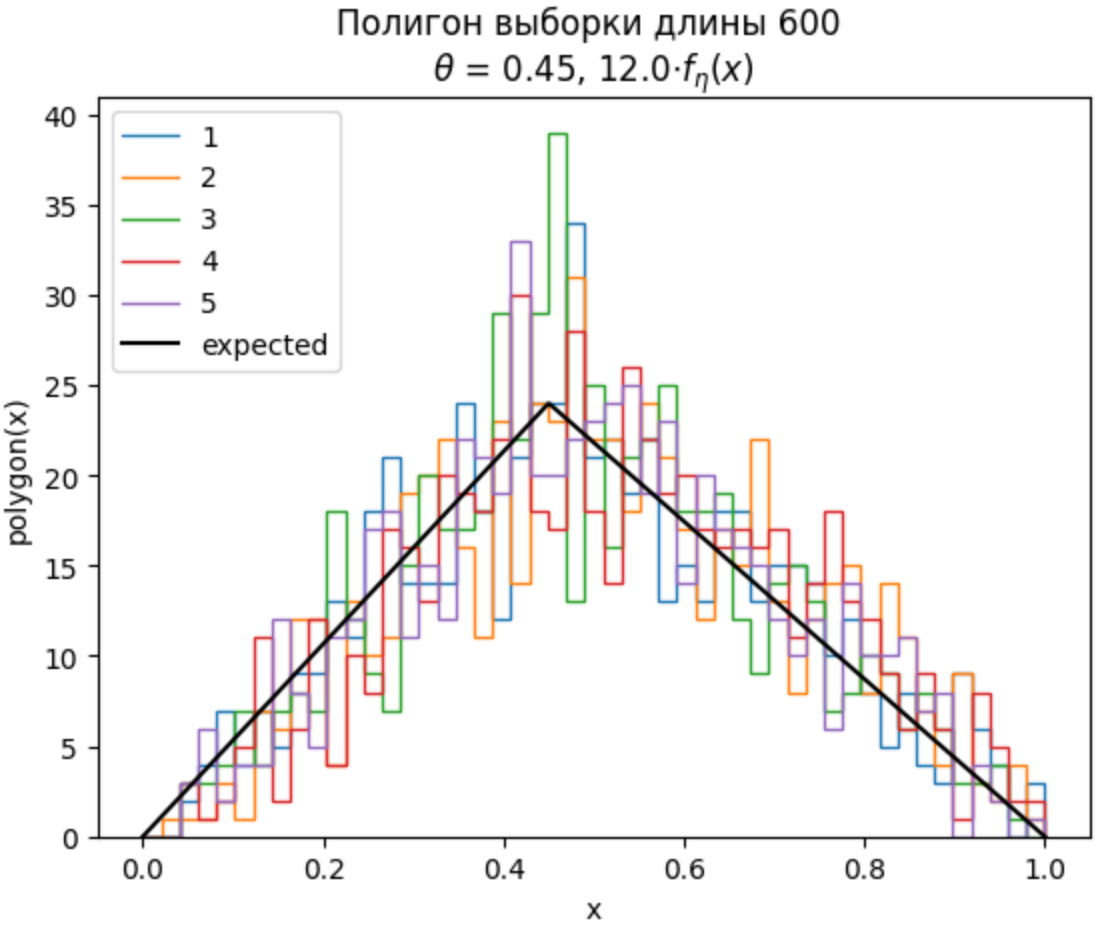
In [34]:
```
# n=200
for i in range(5):
    plt.stairs(Y_poleta[3][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*200/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 200 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(200/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



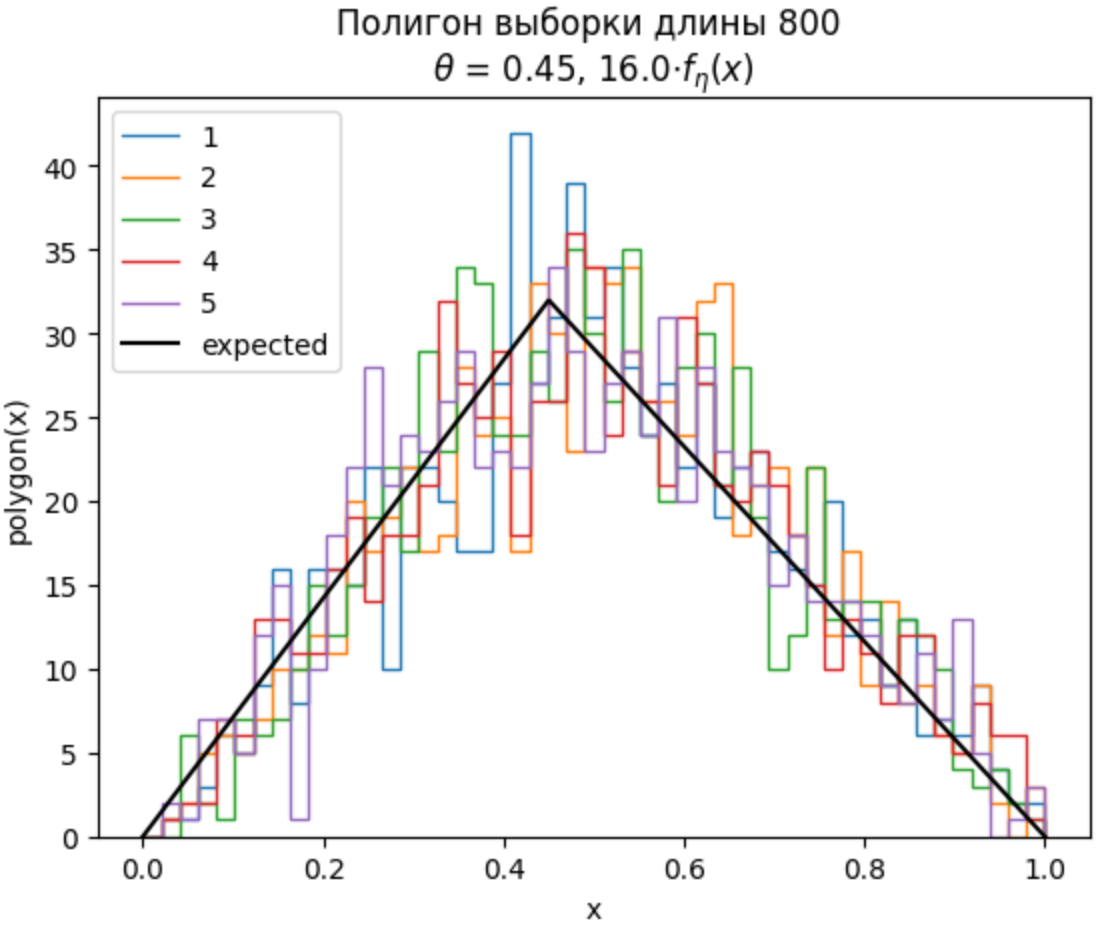
In [35]:
```
# n=400
for i in range(5):
    plt.stairs(Y_poleta[4][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*400/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 400 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(400/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



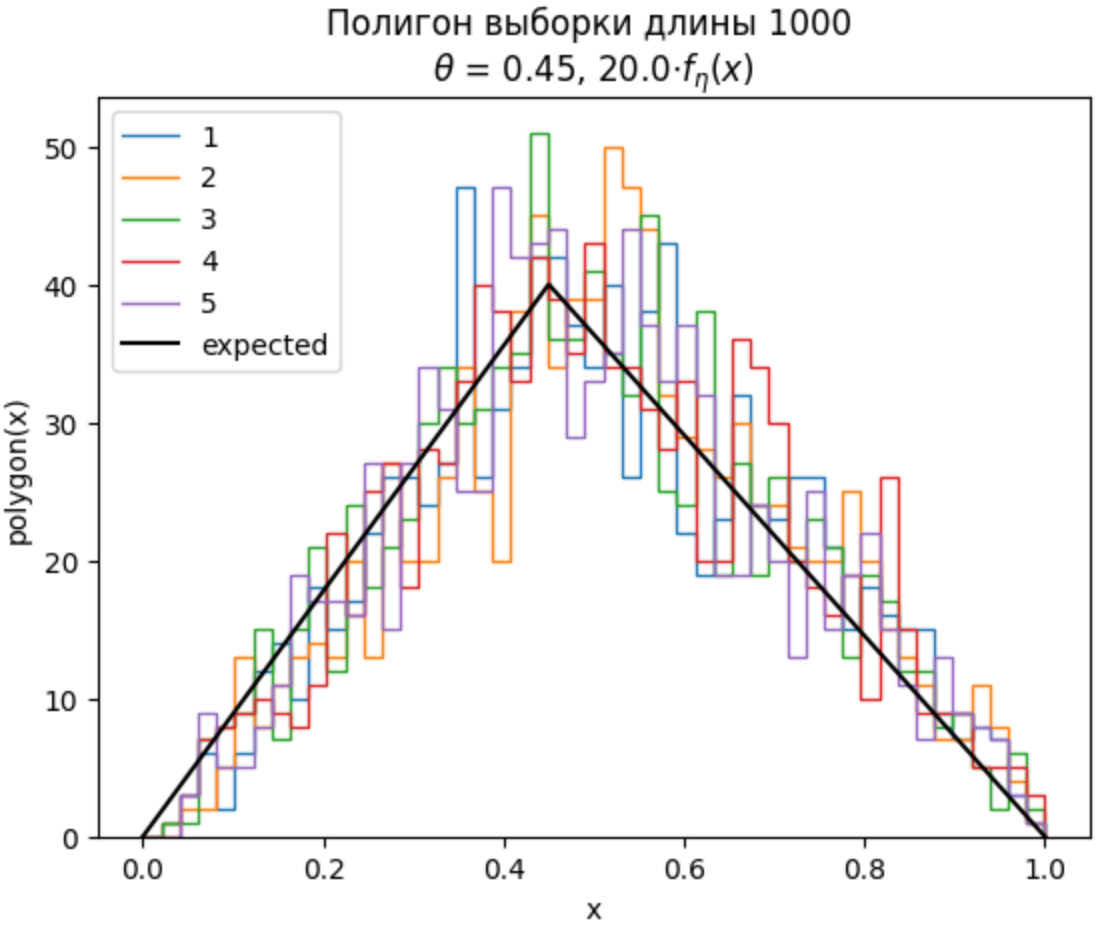
In [36]:
```
# n=600
for i in range(5):
    plt.stairs(Y_poleta[5][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*600/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 600 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(600/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



In [37]:
```
# n=800
for i in range(5):
    plt.stairs(Y_poleta[6][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*800/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 800 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(800/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



In [38]:
```
# n=1000
for i in range(5):
    plt.stairs(Y_poleta[7][i], np.append(X_poleta, 1))
plt.plot(X_realeta, posibilityeta*1000/50, 'k')
plt.xlabel('x')
plt.ylabel('polygon(x)')
plt.title('Полигон выборки длины 1000 \n$\\theta$ = 0.45, ' +
          '%.1f$\\cdot f_{\\eta}(x)$'%(1000/50))
plt.legend([1, 2, 3, 4, 5, 'expected'], loc='upper left');
```



Чтобы отметить, что я все еще анализирую графики, а не просто переписываю код под непрерывный случай: графики и результаты двух предыдущих заданий опять демонстрируют справедливость теоремы.

Задание 4

In [39]:

```
def eta_sample_mean(sample):
    return sum(sample)/len(sample)

def eta_sample_variance(sample):
    return sum((sample-xi_sample_mean(sample))**2)/len(sample)

meanseta = np.array([[eta_sample_mean(sample_eta[k][j])for j in range(5)]
                     for k in range(len(n))])
varianceseta = np.array([[eta_sample_variance(sample_eta[k][j])for j in range(5)]
                         for k in range(len(n))])
means_peta = np.array([['%.4f' % i for i in j] for j in meanseta])
variances_peta = np.array([['%.4f' % i for i in j] for j in varianceseta])
expectationeta = (1+0.45)/3
varianceeta = (1-0.45+0.45**2)/18
means_difeta = np.array([['%.4f' % i for i in j]
                         for j in (meanseta-expectationeta)])
variances_difeta = np.array([['%.4f' % i for i in j]
                             for j in (varianceseta-varianceeta)])
```

In [40]:
```
fig, ax = plt.subplots(2,1, figsize=(7,6.5))
ax[0].table(cellText = means_peta, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[0].set_axis_off()
ax[0].set_title('Выборочные средние');
ax[1].table(cellText = variances_peta, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title('Выборочные дисперсии');
```

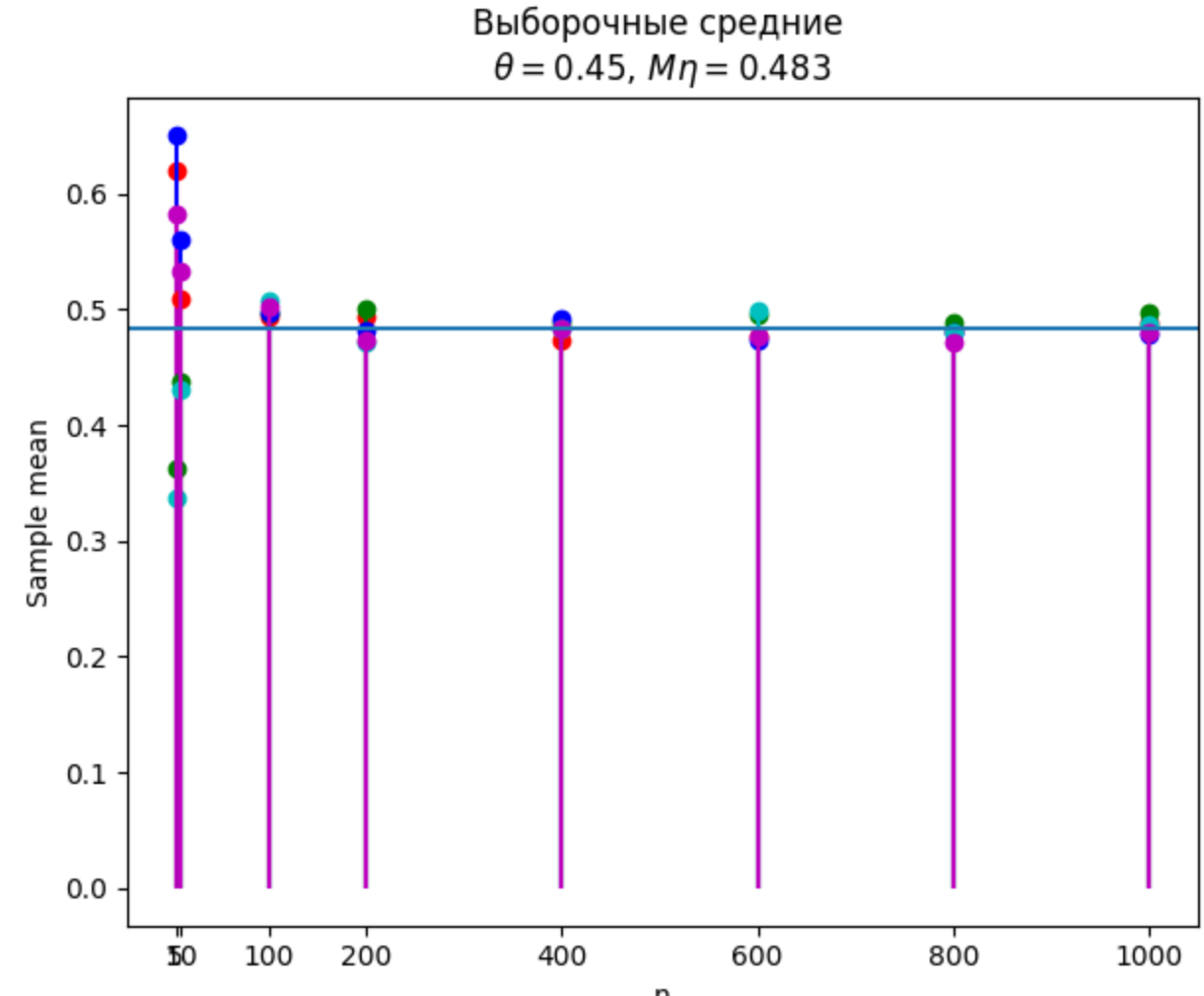
Выборочные средние

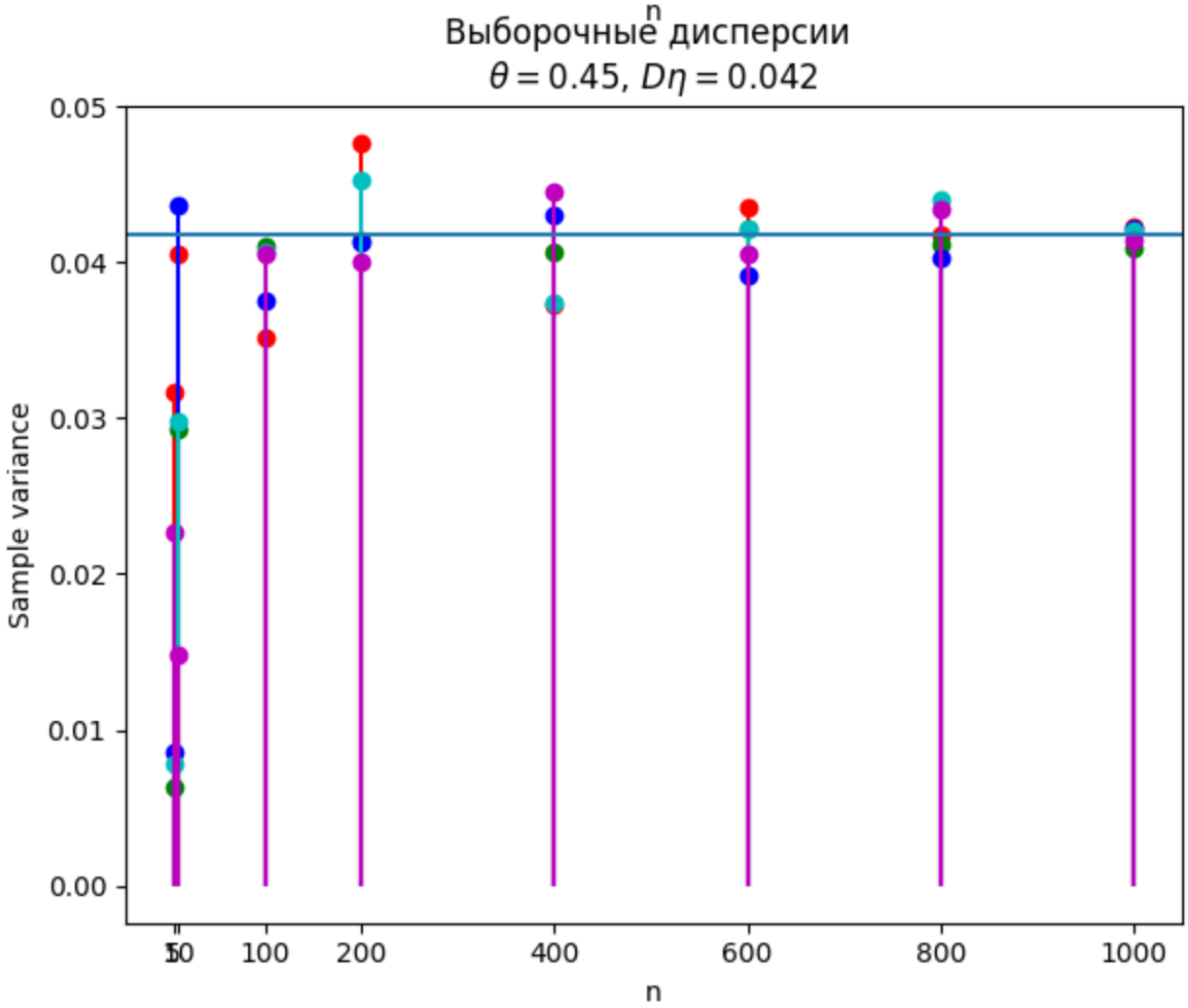
	1	2	3	4	5
5	0.6194	0.3616	0.6509	0.3366	0.5829
10	0.5088	0.4373	0.5594	0.4300	0.5335
100	0.4935	0.4977	0.4969	0.5071	0.5024
200	0.4934	0.5006	0.4812	0.4724	0.4741
400	0.4725	0.4889	0.4927	0.4838	0.4831
600	0.4759	0.4949	0.4733	0.4983	0.4765
800	0.4823	0.4887	0.4805	0.4809	0.4710
1000	0.4881	0.4978	0.4785	0.4863	0.4797

Выборочные дисперсии

	1	2	3	4	5
5	0.0317	0.0064	0.0086	0.0078	0.0227
10	0.0405	0.0293	0.0436	0.0297	0.0148
100	0.0351	0.0410	0.0375	0.0406	0.0406
200	0.0477	0.0414	0.0413	0.0453	0.0400
400	0.0372	0.0406	0.0431	0.0374	0.0446
600	0.0435	0.0422	0.0392	0.0422	0.0405
800	0.0417	0.0412	0.0403	0.0440	0.0434
1000	0.0423	0.0409	0.0421	0.0420	0.0414

In [41]:
```
fig, ax = plt.subplots(2,1, figsize=(7,12))
for k in range(len(n)):
    for j in range(5):
        ax[0].stem(n[k], meanseta[k][j], 'rgbcm'[j])
ax[0].axhline(expectationeta)
ax[0].set(xticks = n, xlabel = 'n', ylabel = 'Sample mean',
          title = 'Выборочные средние \n$\\theta = 0.45, ' +
                  '\\,M\\eta = %.3f$'%expectationeta)

for k in range(len(n)):
    for j in range(5):
        ax[1].stem(n[k], varianceseta[k][j], 'rgbcm'[j])
ax[1].axhline(y = varianceeta)
ax[1].set(xticks = n, xlabel = 'n', ylabel = 'Sample variance',
          title = 'Выборочные дисперсии \n$\\theta = 0.45, ' +
                  '\\,D\\eta = %.3f$'%varianceeta);
```


Выборочные средние
$\theta = 0.45$, $M\eta = 0.483$
Sample mean
0.0
0.1
0.2
0.3
0.4
0.5
0.6
10
100
200
400
600
800
1000
n


Выборочные дисперсии
$\theta = 0.45$, $D\eta = 0.042$
Sample variance
0.00
0.01
0.02
0.03
0.04
0.05
10
100
200
400
600
800
1000
n

In [42]:
```
fig, ax = plt.subplots(2,1, figsize=(7,7))
ax[0].table(cellText = means_difeta, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[0].set_axis_off()
ax[0].set_title('Смещение оценки: $b(\\theta)=M_{\\theta}T(x)-\\tau(\\theta)$' +
                '\n $M\\eta = %.4f$'%expectationeta);

ax[1].table(cellText = variances_difeta, rowLabels=n, colLabels=[1,2,3,4,5],
            loc='center').scale(1, 1.5)
ax[1].set_axis_off()
ax[1].set_title('Разница выборочной дисперсии и дисперсии' +
                '\n $D\\eta = %.4f$'%varianceeta);
```

Смещение оценки: $b(\theta) = M_\theta T(x) - \tau(\theta)$
$M\eta = 0.4833$

	1	2	3	4	5
5	0.1361	-0.1217	0.1675	-0.1467	0.0996
10	0.0254	-0.0461	0.0760	-0.0534	0.0501
100	0.0101	0.0144	0.0136	0.0238	0.0191
200	0.0101	0.0173	-0.0021	-0.0110	-0.0092
400	-0.0108	0.0056	0.0093	0.0005	-0.0003
600	-0.0075	0.0115	-0.0100	0.0149	-0.0069
800	-0.0010	0.0053	-0.0028	-0.0025	-0.0123
1000	0.0047	0.0145	-0.0048	0.0029	-0.0036

Разница выборочной дисперсии и дисперсии
$D\eta = 0.0418$

	1	2	3	4	5
5	-0.0101	-0.0354	-0.0332	-0.0340	-0.0191
10	-0.0013	-0.0125	0.0018	-0.0121	-0.0270
100	-0.0067	-0.0008	-0.0043	-0.0012	-0.0012
200	0.0058	-0.0004	-0.0005	0.0035	-0.0018
400	-0.0046	-0.0012	0.0013	-0.0044	0.0028
600	0.0017	0.0004	-0.0026	0.0004	-0.0013
800	-0.0001	-0.0006	-0.0015	0.0022	0.0016
1000	0.0005	-0.0009	0.0003	0.0002	-0.0004

Смещение оценки уменьшается с увеличением числа элементов выборки, что говорит о состоятельности оценки

... - и что еще раз на практике подтверждает теорему - ...

In [ ]: