

Свойства статистик. Исследование свойств различных оценок параметров распределений. Несмешенность и состоятельность статистик

монета | **МАТБАЗА**

**Свойства статистик. Исследование свойств оценок параметров распределений. Информация Фишера и оценка Рао-Крамера.**

**22 ноября**  
занятие 2.5.



**Игорь Нехаев**  
Специалист по анализу данных и машинному обучению

▼ Изучаем распределения разных оценок для генерального среднего

Пусть у нас есть выборка СВ "Время подключения клиентов по заявкам в 2023 году" объемом  $n$ :  $\bar{X} = (X_1, X_2, \dots, X_n)$ .

Какая статистика будет лучше для оценки генерального среднего?

Рассмотрим две статистики:

- выборочное среднее, она нам хорошо известна

$$\bar{x}_{\text{выб}} = \frac{X_1 + X_2 + \dots + X_n}{n}$$

- среднее арифметическое минимума и максимума выборки:

$$\bar{x}_{\text{ММ}} = \frac{\min(\bar{X}) + \max(\bar{X})}{2}$$

Забегая вперед скажу, что далее мы выведем свойства оценок и докажем, что выборочное среднее - наилучшая оценка генерального среднего из всех.

А сейчас мы ограничимся экспериментом - построим эмпирические распределения данных оценок и сравним их.

**Описание эксперимента**

Чтобы построить закон распределения оценки (статистики) надо:

1. провести много наблюдений ее значений. Одно наблюдение состоит в следующем:

- получение выборки объема  $n$  из данной ГС:  $(x_1, \dots, x_n)$  (значение векторной СВ  $\bar{X}$ );
- вычисление значения статистики №1 и статистики №2.

2. Проведя  $N$  (много) таких наблюдений, мы сможем построить эмпирические гистограммы распределения оценок (оценить  $h_i$ ), выбрав границы носителя и шаг (кол-во шагов).

3. Исследуем свойства распределений и определим какая оценка лучше



## ВОПРОС

Допустим, что мы знаем генеральное среднее  $m_X$  и построили два эмпирических распределения оценок на основе большого количества наблюдений. И нашли мат.ожидание каждой оценки с помощью полученных эмпирических распределений соответственно:  $m_{\bar{x}_{\text{байб}}}$ ,  $m_{\bar{x}_{\text{ММ}}}$ . Также нашли и их дисперсии:  $D_{\bar{x}_{\text{байб}}}$ ,  $D_{\bar{x}_{\text{ММ}}}$ . Сравнивая эти характеристики, определите какая из оценок лучше. Выберите наиболее точный с вашей точки зрения ответ:

Лучше  $\bar{x}_{\text{байб}}$ , если ...

1.  $(m_{\bar{x}_{\text{байб}}} < m_{\bar{x}_{\text{ММ}}}) \text{ и } (D_{\bar{x}_{\text{байб}}} < D_{\bar{x}_{\text{ММ}}})$ ;
2.  $(m_{\bar{x}_{\text{байб}}} > m_{\bar{x}_{\text{ММ}}}) \text{ и } (D_{\bar{x}_{\text{байб}}} > D_{\bar{x}_{\text{ММ}}})$ ;
3.  $(|m_{\bar{x}_{\text{байб}}} - m_X| < |m_{\bar{x}_{\text{ММ}}} - m_X|)$ ;
4.  $(|m_{\bar{x}_{\text{байб}}} - m_X| < |m_{\bar{x}_{\text{ММ}}} - m_X|) \text{ и } (D_{\bar{x}_{\text{байб}}} < D_{\bar{x}_{\text{ММ}}})$ .

## ▼ Ставим эксперимент

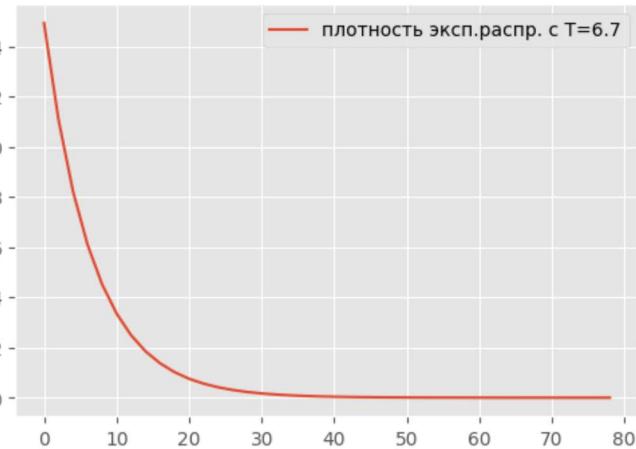
В качестве СВ X рассмотрим

- "Время подключения заявки", подчиняющуюся экспоненциальному закону распределения с параметром (генеральным средним)  $T_{\text{mean}}=6.7$ ;
- "Сумма выручки за день продажи кофемашин" (см. урок №7);

```
# подгружаем нужные библиотеки и устанавливаем нужные значения параметров
import numpy as np
np.set_printoptions(precision=3)
import scipy.stats as stat
import matplotlib.pyplot as plt
plt.style.use('ggplot')
```

```
# создадим экспоненциальную распределенную СВ "Время подключения заявки"
Tmean = 6.7
exp_val = stat.expon(scale=Tmean)

# нарисуем плотность распределения интервала времени
xbins = np.arange(0, 80, 2)
plt.figure(figsize=(6, 4))
plt.plot(xbins, exp_val.pdf(xbins), label=f"плотность эксп.распр. с T={Tmean}")
plt.legend()
plt.show()
```



```
# создадим нормально распределенную СВ "Сумма выручки за день продажи кофемашин" (см. урок №7)
mu = 875; sigma = 146
norm_rv = stat.norm(loc=mu, scale=sigma)

# --- построим график плотности распределения
# сгенерируем 100 значений
xvals = norm_rv.rvs(size=100)
# вычислим значения функции плотности вероятности нормального распределения
x_diapason = xvals.max() - xvals.min()
x = np.linspace(xvals.min() - 0.1 * x_diapason, xvals.max() + 0.1 * x_diapason, 100)
pdf = norm_rv.pdf(x)

plt.plot(x, pdf, label = 'плотность распределения' )
plt.ylabel('$\phi(x, m, \sigma)$'); plt.xlabel('$x$')
plt.legend()
plt.suptitle(f'Рис. Плотность нормального распределения с параметрами $m={mu}$, $\sigma={sigma}$', y=0);
```

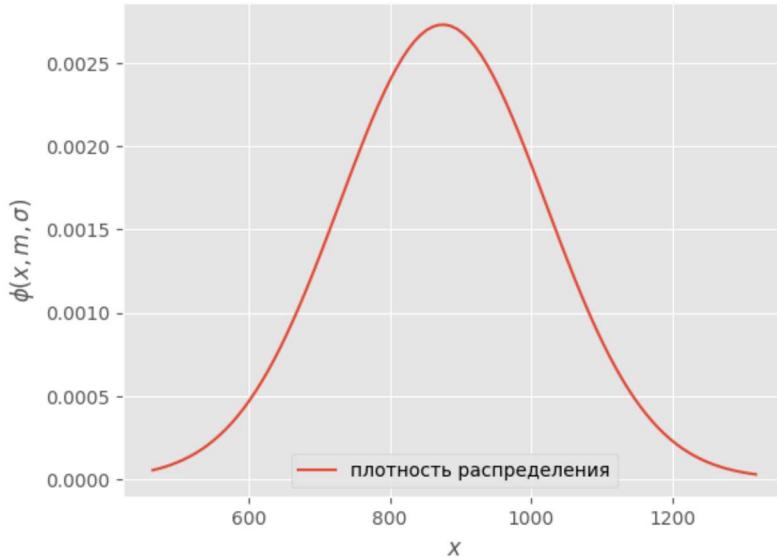


Рис. Плотность нормального распределения с параметрами  $m = 875, \sigma = 146$

ЭТАП 1. Провести много наблюдений значений статистик:

- генерируем выборки объема  $n$  из данной ГС:  $(x_1, \dots, x_n)$  (значение векторной СВ  $\bar{X}$ );
- вычисляем значения статистики №1 и статистики №2 на каждой выборке.

```
def estimate_sample(rand_val, estim_fun, sample_size, N=1000):
    samples = [rand_val.rvs(size=n) for _ in range(N)]
    estims = np.array([estim_fun(X) for X in samples])
    return estims

stat2 = lambda X: (np.min(X)+ np.max(X))/2
stat3 = lambda X: (2*sum(X) - np.min(X) - np.max(X)) / (2*len(X)-2)

n = 10 # объем выборки
N = 1000 # кол-во наблюдений/выборок

#rand_val = exp_val # выбираем экспоненциально распределенную СВ
rand_val = norm_rv # выбираем нормально распределенную СВ
if rand_val == exp_val:
    m_X = Tmean
else:
    m_X = mu

estim1 = estimate_sample(rand_val=rand_val, estim_fun=np.mean, sample_size=n, N=N)
estim2 = estimate_sample(rand_val=rand_val, estim_fun=stat2, sample_size=n, N=N)
print(f"интервал изменения статистики №1: {min(estim1)} - {max(estim1)}")
print(f"интервал изменения статистики №2: {min(estim2)} - {max(estim2)}")

интервал изменения статистики №1: 722.8861309883712 - 1004.628530433122
интервал изменения статистики №2: 565.4665364940249 - 1058.3818980547785
```

ЭТАП 2. Построим эмпирические гистограммы распределения оценок (оценим  $h_i$ ), выбрав границы носителя и шаг (кол-во шагов)

```

if rand_val == exp_val:
    # определить интервал изменения значений статистик
    x_min, x_max = 0, 37
    # разбить его на n подинтервалов
    nn = 37
    x_i = np.linspace(x_min, x_max, nn+1)
else:
    # определить интервал изменения значений статистик
    x_min, x_max = 600, 1100
    # разбить его на nn подинтервалов
    nn = 26
    x_i = np.linspace(x_min, x_max, nn+1)

x_i

array([ 600.   ,  619.231,  638.462,  657.692,  676.923,  696.154,
       715.385,  734.615,  753.846,  773.077,  792.308,  811.538,
       830.769,  850.   ,  869.231,  888.462,  907.692,  926.923,
       946.154,  965.385,  984.615, 1003.846, 1023.077, 1042.308,
      1061.538, 1080.769, 1100.   ])

```

```

def gen_rv_distrib_fun(vals: np.ndarray, xbins=None):
    if xbins is None:
        nbins = 20
        x_order = int(np.log10(vals.max() - vals.min()))
        x_min = round(vals.min() / 10**x_order - 0.5) * 10**x_order
        x_max = round(vals.max() / 10**x_order + 0.5) * 10**x_order
        xbins = x_min + np.arange(0, nbins + 1) * (x_max - x_min) / nbins
    h_i, x_i, _ = plt.hist(vals, bins=xbins, density=True, alpha=0.4)

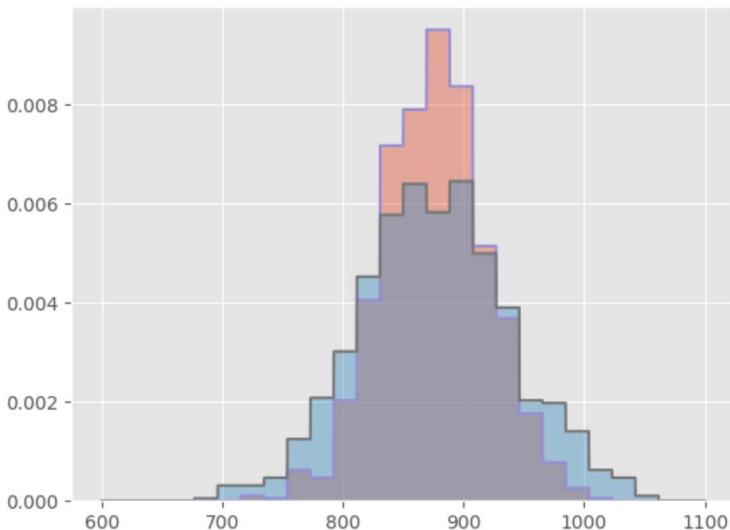
def rv_distrib_fun(x, h_i=h_i, x_i=x_i):
    if x_i[0] <= x <= x_i[-1]:
        i = int(len(h_i) * (x - x_i[0]) / (x_i[-1] - x_i[0]))
        #print(x, i)
        if i == len(h_i):
            i -= 1
        while not(x_i[i] <= x <= x_i[i+1]):
            if x < x_i[i]:
                i -= 1
            else:
                i += 1
        return h_i[i]
    else:
        return 0
    return rv_distrib_fun

```

```

estim1_fun = gen_rv_distrib_fun(estim1, x_i)
estim2_fun = gen_rv_distrib_fun(estim2, x_i)
x_arr = np.arange(x_i[0], x_i[-1], 0.1)
y1 = [estim1_fun(x) for x in x_arr]
y2 = [estim2_fun(x) for x in x_arr]
plt.plot(x_arr, y1)
plt.plot(x_arr, y2)
plt.show()

```



ЭТАП 3. Исследуем свойства распределений и определим какая оценка лучше

Для этого вычислим среднее и дисперсию для данных оценок с использованием полученных эмпирических распределений

```

from scipy import integrate
import scipy.stats as stat

# зададим подинтегральные функции
f1_X = lambda x: estim1_fun(x) * x
f2_X = lambda x: estim2_fun(x) * x

# найдем M(X) - выполним интегрирование с помощью пакета
m1 = round(integrate.quad(f1_X, min(x_i), max(x_i))[0], 2)
m2 = round(integrate.quad(f2_X, min(x_i), max(x_i))[0], 2)

print(f"M[estim1] = {m1}, отклонение от истинного= {round(m1 - m_X, 3)}")
print(f"M[estim2] = {m2}, отклонение от истинного= {round(m2 - m_X, 3)}")

M[estim1] = 876.26, отклонение от истинного= 1.26
M[estim2] = 876.66, отклонение от истинного= 1.66
<ipython-input-48-0d598ad6e566>:9: IntegrationWarning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties. If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges. Perhaps a special-purpose integrator should be used.
m1 = round(integrate.quad(f1_X, min(x_i), max(x_i))[0], 2)
<ipython-input-48-0d598ad6e566>:10: IntegrationWarning: The maximum number of subdivisions (50) has been achieved.
  If increasing the limit yields no improvement it is advised to analyze
  the integrand in order to determine the difficulties. If the position of a
  local difficulty can be determined (singularity, discontinuity) one will
  probably gain from splitting up the interval and calling the integrator
  on the subranges. Perhaps a special-purpose integrator should be used.
m2 = round(integrate.quad(f2_X, min(x_i), max(x_i))[0], 2)

# зададим подинтегральные функции
f1_X2 = lambda x: estim1_fun(x) * x**2
f2_X2 = lambda x: estim2_fun(x) * x**2

# найдем M(X) - выполним интегрирование с помощью пакета
m1_X2 = round(integrate.quad_vec(f1_X2, min(x_i), max(x_i))[0], 2)
m2_X2 = round(integrate.quad_vec(f2_X2, min(x_i), max(x_i))[0], 2)
D1 = m1_X2 - m1**2
D2 = m2_X2 - m2**2
sigma1, sigma2 = np.sqrt(D1), np.sqrt(D2)
print(f"D[estim1] = {round(D1, 2)}, CKO = {round(sigma1, 2)}")
print(f"D[estim2] = {round(D2, 2)}, CKO = {round(sigma2, 2)}")

D[estim1] = 1831.18, CKO = 42.79
D[estim2] = 3261.65, CKO = 57.11

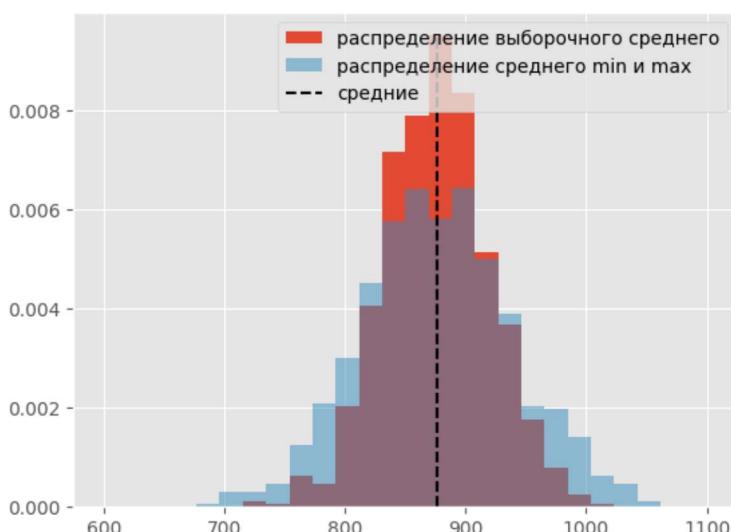
```

собственно, мы этого и ожидали, ведь более лучшая оценка должна в среднем давать истинное значение параметра (быть несмещенной) и в среднем меньше от него отклоняться.

```

plt.hist(estim1, bins=x_i, density=True, label='распределение выборочного среднего')
plt.hist(estim2, bins=x_i, density=True, alpha=0.5, label='распределение среднего min и max')
plt.vlines([m1, m2], ymin=0, ymax=[np.max(y1), np.max(y2)], colors='black', linestyles='--', label='средние')
plt.legend()
plt.show()

```



Для оценки среднего СВ, подчиняющейся экспоненциальному закону распределения

при n=10 получили:

- M[estim1] = 6.66, отклонение от истинного= -0.04;
- M[estim2] = 9.94, отклонение от истинного= 3.24;
- D[estim1] = 4.41, СКО = 2.1
- D[estim2] = 15.35, СКО = 3.92

при n=20 получили:

- M[estim1] = 6.68, отклонение от истинного= -0.02
- M[estim2] = 12.53, отклонение от истинного= 5.83
- D[estim1] = 2.2, СКО = 1.48
- D[estim2] = 18.92, СКО = 4.35

Для оценки среднего СВ, подчиняющейся нормальному закону распределения

при n=10 получили:

- M[estim1] = 17537.5, отклонение от истинного= 7.3
- M[estim2] = 17508.45, отклонение от истинного= -21.75
- D[estim1] = 1028.0, СКО = 32.06
- D[estim2] = 38136.98, СКО = 195.29

при n=20 получили:

- M[estim1] = 17537.92, отклонение от истинного= 1.12
- M[estim2] = 17549.95, отклонение от истинного= 13.15



## ▼ ЗАДАНИЕ. Проверяем интуицию

Пусть при оценке генерального среднего для СВ "Время подключения клиента по заявке" мы использовали такие две оценки:

- выборочное среднее

$$x_{cp}(n) = \frac{X_1 + X_2 + \dots + X_n}{n}$$

- выборочное средневзвешенное, в котором веса крайних значений выборки ( $x_{min}$  и  $x_{max}$ ) в два раза меньше весов остальных значений выборки:

$$x_{B3B}(n) = \frac{2 * (X_1 + X_2 + \dots + X_n) - max(X) - min(X)}{2n - 2}$$

Попробуйте спрогнозировать какими свойствами будет обладать средневзвешенная оценка:

ВАРИАНТЫ ОТВЕТОВ:

1.  $m_{x_{B3B}(n)}$  будет точнее, чем  $m_{x_{cp}(n)}$  и  $D_{x_{B3B}(n)} \leq D_{x_{cp}(n)}$  при любых n
2.  $m_{x_{B3B}(n)}$  будет точнее, чем  $m_{x_{cp}(n)}$ , но  $D_{x_{B3B}(n)} \geq D_{x_{cp}(n)}$  при любых n
3.  $m_{x_{B3B}(n)}$  будет сильнее отклоняться от Ген.Ср., чем  $m_{x_{cp}(n)}$ , но  $D_{x_{B3B}(n)} \leq D_{x_{cp}(n)}$  при любых n
4.  $m_{x_{B3B}(n)}$  будет сильнее отклоняться от Ген.Ср., чем  $m_{x_{cp}(n)}$  и  $D_{x_{B3B}(n)} \geq D_{x_{cp}(n)}$  при любых n

Итак, мы теперь понимаем какая оценка нам нужна - та, у которой закон распределения "жмется" к истинному значению параметра.

## ▼ Свойства статистик

Рассмотрим определения важных свойств оценок параметров распределений.

Пусть СВ  $X$  подчиняется некоторому закону распределения с параметром  $\Theta$  и плотностью  $f_X(x, \Theta)$ , зависящем от данного параметра.

Например, для экспоненциального распределения  $\Theta = T_{cp}$  и  $f_X(x, T_{cp}) = \frac{1}{T_{cp}} \cdot e^{-x/T_{cp}}$

В более общем случае, у плотности может быть несколько параметров, тогда параметр  $\Theta$  можно считать вектором параметров.

И пусть мы имеем некоторую выборку размером  $n$ :

$$\bar{X}_n = (X_1, \dots, X_n).$$

На основании этой выборки мы вычисляем некоторую статистику, которая является оценкой параметра  $\Theta$ :

$$\hat{\Theta}_n = f(\bar{X}_n)$$

Данная статистика является СВ и подчиняется некоторому закону распределения и имеет такие числовые характеристики как  $m_{\hat{\Theta}_n}$  и  $D_{\hat{\Theta}_n}$ .

Рассмотрим основные свойства такой оценки.

## ▼ Несмешенность оценки

---

**Опр.** Оценка параметра  $\hat{\Theta}_n$  называется **несмешенной**, если  $\forall n$  выполняется:

$$M[\hat{\Theta}_n] = \Theta$$

Наше исследование показало, что оценка "выборочное среднее" является несмешенной. Отклонение  $|[\hat{\Theta}_n]_{cp} - \Theta|$  очень мало при  $N=1000$ . Если же увеличить  $N$  (кол-во наблюдений статистики), то и отклонение уменьшится независимо от размера выборки  $n$ .

Несмешенность  $x_{cp}(n)$  можно легко доказать:

$$M[x_{cp}(n)] = M\left[\frac{X_1 + X_2 + \dots + X_n}{n}\right] = \frac{M[X_1] + M[X_2] + \dots + M[X_n]}{n} = \frac{n \cdot m_X}{n} = m_X$$

Проанализируем на несмешенность **выборочную дисперсию**.

Случай 1. Нам известно генеральное среднее  $m_X$

$$M[S_X^2(n)] = M\left[\frac{(X_1 - m_X)^2 + \dots + (X_n - m_X)^2}{n}\right] = \frac{M[(X_1 - m_X)^2] + \dots + M[(X_n - m_X)^2]}{n} = \frac{n \cdot D_X}{n} = D_X$$

Здесь док-во оказалось очень простым. Рассмотрим 2-й случай, когда вместо  $m_X$  мы используем  $x_{cp}(n)$

Случай 2. Нам неизвестно генеральное среднее  $m_X$  и мы используем выборочное среднее  $x_{cp}(n)$

$$M[S_X^2(n)] = M\left[\frac{(X_1 - x_{cp}(n))^2 + \dots + (X_n - x_{cp}(n))^2}{n}\right] = \frac{M[(X_1 - x_{cp}(n))^2] + \dots + M[(X_n - x_{cp}(n))^2]}{n}$$

Рассмотрим чему равно одно слагаемое в числителе

$$M[(X_i - x_{cp}(n))^2] = M[X_i^2 - 2X_i \cdot x_{cp}(n) + x_{cp}(n)^2] = m_{X^2} - 2M[X_i \cdot \frac{X_1 + X_2 + \dots + X_n}{n}] + M[(\frac{X_1 + X_2 + \dots + X_n}{n})^2]$$

С учетом того, что

- $M[X_i \cdot X_j] = M[X_i] \cdot M[X_j] = m_X^2, i \neq j;$
- $M[X_i \cdot X_i] = m_{X^2}$

получаем после всех преобразований и сокращений (более подобный вывод далее, при анализе состоятельности):

$$\begin{aligned} M[(X_i - x_{cp}(n))^2] &= m_{X^2} - 2M[X_i \cdot \frac{X_1 + X_2 + \dots + X_n}{n}] + M[(\frac{X_1 + X_2 + \dots + X_n}{n})^2] = m_{X^2} - \frac{1}{n} \cdot (m_{X^2} + (n-1)m_X^2) \\ &= \frac{n-1}{n} \cdot (m_{X^2} - m_X^2) = \frac{n-1}{n} \cdot D_X \end{aligned}$$

Имеем:

$$M[S_X^2(n)] = \frac{n-1}{n} \cdot D_X = D_X - \frac{1}{n} \cdot D_X$$

Таким образом, видим, что эта оценка занижает дисперсию (подумайте почему?) и смещение составляет  $\frac{1}{n} \cdot D_X$ . С ростом  $n$  это смещение стремится к нулю и такая оценка называется асимптотически несмешенной.

На практике, при небольших  $n$  лучше использовать **несмешенную оценку** для дисперсии:

$$S_X^2(n-1) = \frac{n}{n-1} \cdot S_X^2(n) = \frac{n}{n-1} \cdot \frac{(X_1 - x_{cp}(n))^2 + \dots + (X_n - x_{cp}(n))^2}{n} = \frac{(X_1 - x_{cp}(n))^2 + \dots + (X_n - x_{cp}(n))^2}{n-1}$$

Действительно:

$$M[S_X^2(n-1)] = \frac{n}{n-1} \cdot M[S_X^2(n)] = \frac{n}{n-1} \cdot \frac{n-1}{n} \cdot D_X = D_X$$

Точно также можно доказать, что оценка

$$\bar{x}_{mm} = \frac{\min(\bar{X}) + \max(\bar{X})}{2}$$

является смещенной в случае несимметричного распределения СВ  $X$  (мы исследуем порядковые статистики на практическом занятии).

Забегая вперед укажем, что порядковыми называют статистики, которые оценивают минимумы - максимумы СВ (например, для равномерно распределенной СВ) или квантили распределения.

## ▼ Состоятельность оценки

Мы уже видели, что хорошая оценка ( $x_{cp}(n)$ ) улучшает свои характеристики с ростом  $n$ . Именно об этом и говорит свойство состоятельности.

ОПРОС.

Можно ли сказать, что если мы возьмем две фиксированные выборки с размерами соответственно  $n_1, n_2$ ,  $n_1 < n_2$ , то гарантированно получим более точное значение оценки  $x_{cp}(n_2)$ , вычисленное по 2-й выборке, чем вычисленное по 1-й выборке  $x_{cp}(n_1)$  меньшего размера?

ВАРИАНТЫ ОТВЕТА

1. ДА
2. НЕТ

### Сходимость по вероятности

После ответа на вышеданный вопрос вы поймете зачем нам данное определение.

**Опр.** Говорят что (оценочная) статистика  $f(X_1, \dots, X_n)$  сходится по вероятности к значению  $\vartheta$  с ростом  $n$ , если:

$$\forall \delta > 0, \forall \varepsilon > 0 (\varepsilon < 1) \exists N : p(|f(X_1, \dots, X_n) - \vartheta| > \delta) < \varepsilon, \forall n > N$$

и обозначается так:

$$f(X_1, \dots, X_n) \xrightarrow{p}_{n \rightarrow \infty} \vartheta$$

### Определение состоятельности

**Опр.** Для оценки  $\hat{\Theta}[n]$  параметра  $\Theta$  состоятельность означает, что она сходится по вероятности к истинному значению параметра  $\Theta$  с ростом  $n$  (объема выборки):

$$\hat{\Theta}[n] \xrightarrow{p}_{n \rightarrow \infty} \Theta$$

На языке характеристик оценок её состоятельность означает:

$$\begin{aligned} m_{\hat{\Theta}[n]} &\rightarrow \Theta(n \rightarrow \infty) \\ D_{\hat{\Theta}[n]} &\rightarrow 0(n \rightarrow \infty) \end{aligned}$$

Если же оценка несмещенная, то достаточно убедиться, что ее дисперсия стремится к нулю с ростом  $n$ .

## ▼ ПРИМЕР. Состоятельность оценки $X_{cp}$

Мы уже знаем, что выборочное среднее - несмещенная оценка. Докажем, что ее дисперсия стремится к нулю с ростом  $n$  - размера выборки.

Найдем сначала Мат.Ожидание ее квадрата:

$$\begin{aligned} m_{x_{cp}^2(n)} &= M[(\frac{X_1 + X_2 + \dots + X_n}{n})^2] = \frac{1}{n^2} M[\sum_{i=1}^n X_i^2 + \sum_{i=1}^n \sum_{j=1, j \neq i}^n X_i \cdot X_j] = \frac{1}{n^2} (\sum_{i=1}^n M[X_i^2] + \sum_{i=1}^n \sum_{j=1, j \neq i}^n M[X_i \cdot X_j]) \\ &= \frac{1}{n^2} (n \cdot m_{X^2} + n \cdot (n-1)m_X^2) = \frac{1}{n} (m_{X^2} + (n-1)m_X^2) \end{aligned}$$

Тогда

$$D_{x_{cp}^2(n)} = m_{x_{cp}^2(n)} - m_{x_{cp}(n)}^2 = \frac{1}{n} (m_{X^2} + (n-1)m_X^2) - m_X^2 = \frac{1}{n} (m_{X^2} - m_X^2) = \frac{1}{n} D_X \rightarrow 0 (n \rightarrow \infty)$$

Таким образом, видим, что  $x_{cp}^2(n)$  - несмещенная и состоятельная оценка.

Это можно подтвердить и экспериментально

```

import numpy as np
from scipy import integrate
import scipy.stats as stat

N = 1000 # кол-во наблюдений/выборок

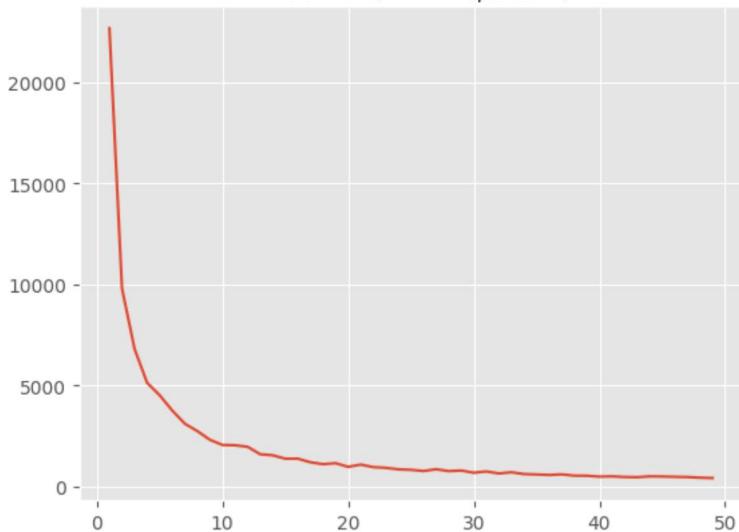
# rand_val = exp_val # выбираем экспоненциально распределенную СВ
rand_val = norm_rv # выбираем нормально распределенную СВ

if rand_val == exp_val:
    m_X = Tmean
else:
    m_X = mu

D_Xcp = []
n_lst = list(range(1, 50))
for n in n_lst:
    estim1 = estimate_sample(rand_val=rand_val, estim_fun=np.mean, sample_size=n, N=N)
    D_Xcp.append(estim1.var())

plt.plot(n_lst, D_Xcp);
plt.title("Снижение дисперсии  $X_{cp}[n]$  с ростом  $n$ ");

```

Снижение дисперсии  $X_{cp}[n]$  с ростом  $n$ 

Посмотрим, как меняется распределение  $X_{cp}$  с ростом  $n$

```

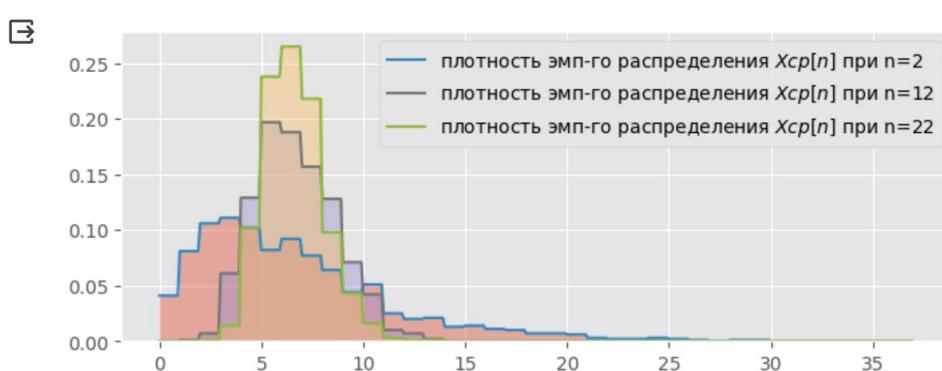
# нарисуем гистограммы эмпирических распределений для  $X_{cp}$  при разных  $n$ 
plt.figure(figsize=(8, 3))
n_lst = list(range(2, 23, 10))
for n in n_lst:
    estim1 = estimate_sample(rand_val=rand_val, estim_fun=np.mean, sample_size=n, N=N)
    estim1_fun = gen_rv_distrib_fun(estim1, x_i)
    x_arr = np.arange(x_i[0], x_i[-1], 0.1)
    y1 = [estim1_fun(x) for x in x_arr]
    plt.plot(x_arr, y1, label=f"плотность эмп-го распределения  $X_{cp}[n]$  при  $n={n}$ ")

```

```

plt.legend()
plt.show()

```



Видно что с ростом  $n$  и разброс значений уменьшается и распределение напоминает нормальное



## ОПРОС

---

Что вы думаете о состоятельности выборочной оценки дисперсии:

$$S_X^2(n) = \frac{(X_1 - m_X)^2 + \dots + (X_n - m_X)^2}{n}$$

1. состоятельна
2. не состоятельна
3. это зависит от распределения СВ X
4. асимптотически состоятельна

## ▼ Эффективность оценки

---

Итак, мы выяснили, что хорошая оценка параметра должна быть несмещенной и состоятельной. Но стремление дисперсии состоятельной оценки к нулю может быть разным. И, если есть возможность выбирать оценки, то из всех несмешанных и состоятельных надо выбирать ту, у которой дисперсия при фиксированном  $n$  наименьшая (при любых фиксированных  $n$ ).

Т.е., если есть несмешанные и состоятельные оценки одного и того же параметра  $\Theta$ :

$$\hat{\Theta}_n = f(\bar{X}_n)$$

и

$$\hat{\Theta}_n^{(i)} = f_i(\bar{X}_n),$$

то выбрать надо ту оценку, у которой дисперсия наименьшая:

$$D_{\hat{\Theta}_n} \leq D_{\hat{\Theta}_n^{(i)}} \quad \forall n = 1.. \infty, \forall i$$

**Опр.** Оценка  $\hat{\Theta}_n$  параметра  $\Theta$  называется эффективной, если она является несмешанной, состоятельной и имеет наименьшую возможную дисперсию.

Осталось теперь выяснить, а как можно оценить снизу дисперсию оценки параметра? Какая дисперсия является наименьшей при заданном  $n$ ?

## Resume

---

Нас будут интересовать статистики  $\hat{\Theta}_n = f(\bar{X})$ , которые:

- являются несмешанными (или асимптотически несмешанными):  $M[\hat{\theta}_n] = \theta$ ;