5 домашнее задание

Импорт библиотек

In [1]:
```
from dataclasses import dataclass
from typing import Callable

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import chi2, norm, t, mode
from sklearn.utils import resample
```

Настройки графиков

In [2]:
```
sns.set(style="whitegrid")
```

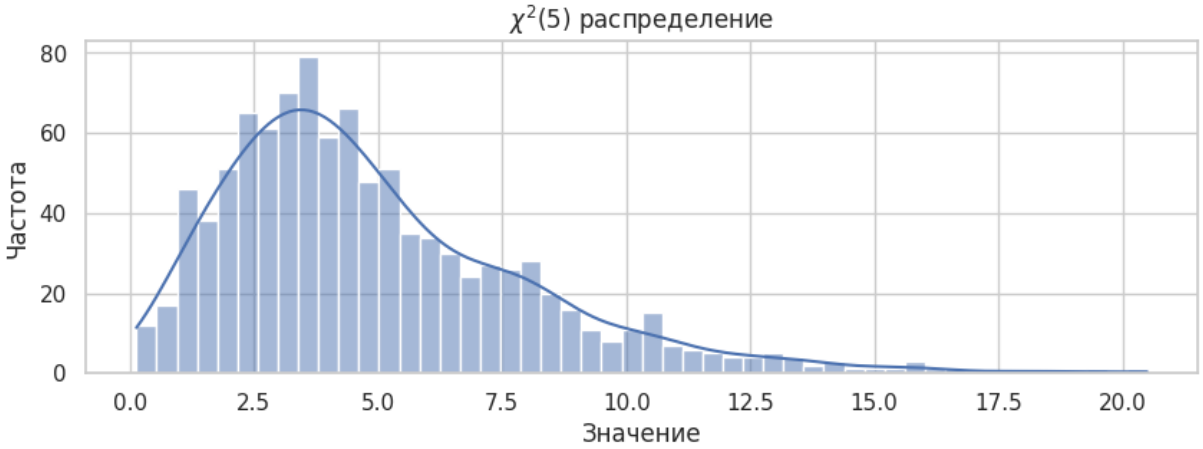
Выбор распределения

Выберем распределение хи-квадрат. Пусть $z_1, \ldots, z_k$ - совместно независимые стандартные нормальные случайные величины, то есть: $z_i \sim N(0, 1)$. Тогда случайная величина $x = z_1 + z_2 + \ldots + z_k$ имеет распределение хи-квадрат с $\mathbf{k}$ **степенями свободы**, то есть $x \sim f_{\chi^2(k)}(x)$, или, если записать по-другому:

$$x = \sum_{i=1}^{k} z_i^2 \sim \chi^2(k)$$

Возьмем готовую функцию для генерации значений из χ^2 распределения из библиотеки **scipy**. Изобразим график при $k = 5$ и $n = 1000$.

In [3]:
```
values = chi2.rvs(5, size=1_000)

plt.figure(figsize=(10, 3))
sns.histplot(values,
             kde=True,
             bins=50)

plt.title('$\chi^2(5)$ распределение')
plt.xlabel('Значение')
plt.ylabel('Частота')

plt.show()
```



Задание №1

Для выбранного распределения сгенерируем выборки разного объема и проиллюстрируем сходимость выборочного среднего к математическому ожиданию.

Нам нужно продемонстрировать центральную предельную теорему. Для примера возьмем $\chi^2(5)$, у которого $\mathbb{E}\chi^2(5) = 5$.

In [4]:
```
df = 5
```

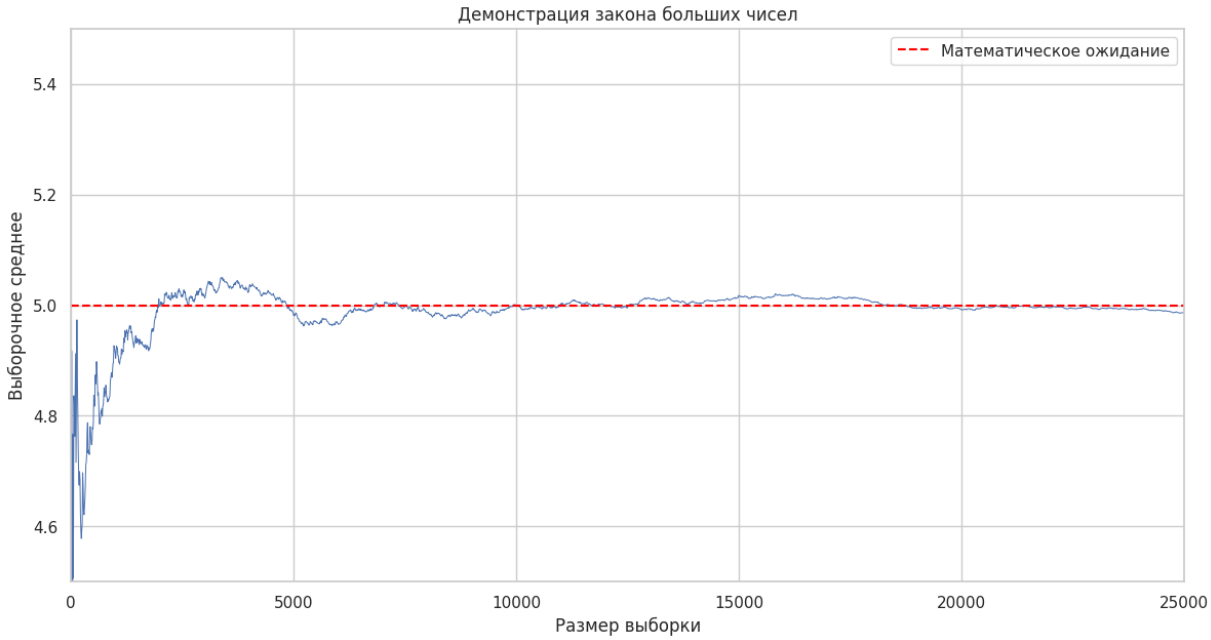
In [5]:
```
n_samples = [5, 20] + [10 ** i for i in range(2, 9)]

data = {'Размер выборки': [],
        'Выборочное среднее': [],
        'Отклонение': []}

for size in n_samples:
    values = chi2.rvs(df, size=size)
    sample_mean = np.mean(values)
    deviation = np.abs(sample_mean - df)

    data['Размер выборки'].append(size)
    data['Выборочное среднее'].append(np.round(np.mean(values), 5))
    data['Отклонение'].append(np.round(np.abs(sample_mean - df), 5))

pd.DataFrame(data)
```

Out[5]:

	Размер выборки	Выборочное среднее	Отклонение
0	5	5.78811	0.78811
1	20	5.15540	0.15540
2	100	4.75596	0.24404
3	1000	5.01009	0.01009
4	10000	5.03014	0.03014
5	100000	5.00256	0.00256
6	1000000	4.99801	0.00199
7	10000000	5.00084	0.00084
8	100000000	5.00007	0.00007

Теперь в виде графика.

In [6]:
```
data = {'Размер выборки': [],
        'Выборочное среднее': []}
n = 25_000
values = chi2.rvs(df, size=n)

for size in range(1, n, 10):
    data['Размер выборки'].append(size)
    data['Выборочное среднее'].append(np.round(np.mean(values[:size]), 5))

plt.figure(figsize=(14, 7))

sns.lineplot(x="Размер выборки",
             y='Выборочное среднее',
             data=pd.DataFrame(data),
             estimator=None,
             linewidth=0.7)

plt.axhline(df,
            label='Математическое ожидание',
            color='red',
            linestyle='dashed')

plt.ylim(df - 0.5, df + 0.5)
plt.xlim(0, n)
plt.legend()
plt.title('Демонстрация закона больших чисел')

plt.show()
```



Задание №2

Наглядно продемонстрируем центральную предельную теорему в действии для хи квадрат распределения.

Нам нужно показать, что:

$$\frac{S_n - \mu n}{\sigma\sqrt{n}} \to N(0,1), \quad n \to \infty$$

Где $S_n = \sum_{i=1}^{n} x_i$ и x_i - последовательность независимых одинаково распределённых случайных величин, имеющих конечное математическое ожидание μ и дисперсию σ^2.

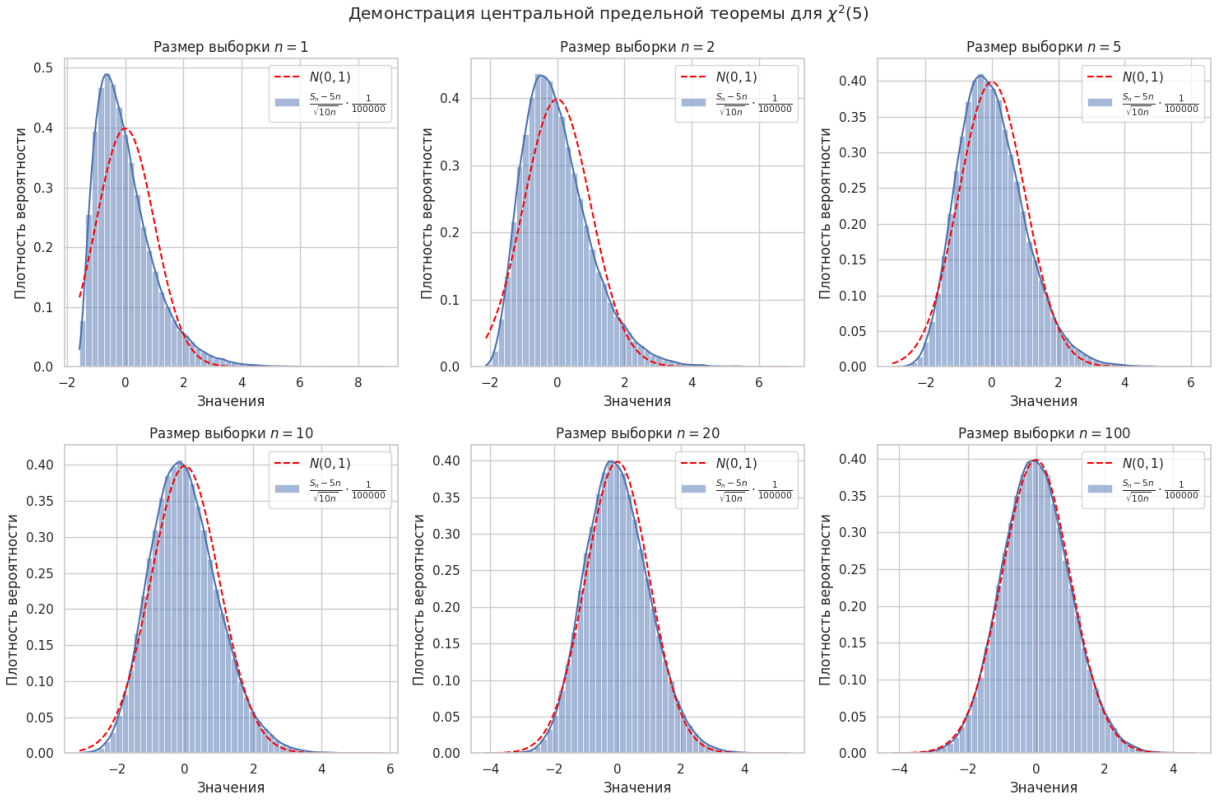
Возьмем $\chi^2(5)$, у которого $\mathbb{E}\chi^2(5) = 5$ и $\mathbb{D}\chi^2(5) = 5 \cdot 2 = 10$, тогда покажем, что:

$$\frac{S_n - 5n}{\sqrt{10n}} \to N(0,1), \quad n \to \infty$$

Где уже $x_i \sim \chi^2(5)$.

In [7]:
```
plt.figure(figsize=(15, 10))

for i, n in enumerate([1, 2, 5, 10, 20, 100], 1):
    plt.subplot(2, 3, i)
    values = chi2.rvs(df, size=(100_000, n)).sum(axis=1)
    values = (values - df * n) / np.sqrt(2 * df * n)
    sns.histplot(values,
```

```
                kde=True,
                bins=50,
                stat='density',
                label='$\\frac{S_{n}-5n}'
                      '{\\sqrt{10n}} \\cdot'
                      '\\frac{1}{100000}$')
    plt.title(f'Размер выборки ${n=}$')
    plt.xlabel('Значения')
    plt.ylabel('Плотность вероятности')

    x = np.linspace(min(values), max(values), 100)
    y = norm.pdf(x, loc=0, scale=1)
    sns.lineplot({'Значения': y, 'Плотность вероятности': x},
                 y='Значения',
                 x='Плотность вероятности',
                 color='red',
                 linestyle='dashed',
                 label='$N(0, 1)$')

plt.suptitle('Демонстрация центральной предельной теоремы для $\chi^2(5)$')
plt.tight_layout()
plt.show()
```



Задание №3

Сгенерируем три выборки $\chi^2(5)$ распределения маленького, среднего и большого размера. Напишем dataclass, чтобы сохранять названия выборок.

In [8]:
```
@dataclass
class Sample:
    values: np.ndarray
    name: str


small_sample = Sample(chi2.rvs(df, size=20), 'маленькая выборка')
medium_sample = Sample(chi2.rvs(df, size=200), 'средняя выборка')
large_sample = Sample(chi2.rvs(df, size=10_000), 'большая выборка')

samples = [small_sample, medium_sample, large_sample]
```

Создадим DataFrame для сохранения результатов

In [9]:
```
# TO-DO
```

Задание №3.1

Напишем функцию для построения асимптотического доверительный интервала для среднего значения на базе ЦПТ.

$$\hat{x} \pm z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$$

Где $z_{1-\frac{\alpha}{2}}$ - это квантиль нормального распределения для заданного уровня значимости. Будем использовать уровень значимости $\alpha = 0.05$. То есть $z_{0.975}$

In [10]:
```
def calc_ci_z(sample: np.ndarray, alpha: float = 0.05) -> tuple:
    z_score = norm.ppf(1 - alpha / 2)

    sample_mean = np.mean(sample)
    sample_std = np.std(sample, ddof=1)

    margin_of_error = z_score * (sample_std / np.sqrt(len(sample)))
    confidence_interval = (sample_mean - margin_of_error,
                           sample_mean + margin_of_error)
    interval_width = 2 * margin_of_error
    return ("Асимтотический доверительный интервал",
           np.round(sample_mean, 4),
           np.round(confidence_interval, 4),
           np.round(interval_width, 4))
```

Для каждой выборки выведем асимптотический доверительный интервал для среднего значения.

Напишем функцию для вывода графиков

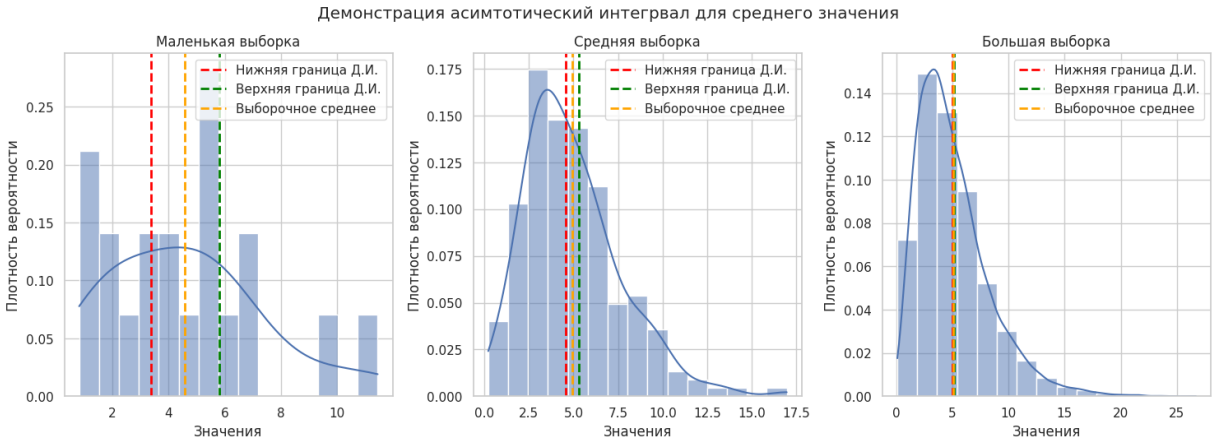
In [11]:
```
def show_graph(samples: list[Sample],
               ci_type: str,
```

```
                stat_func: Callable) -> None:
    plt.figure(figsize=(15, 10))
    plt.suptitle(f'Демонстрация {ci_type}')
    for i, sample in enumerate(samples):
        plt.subplot(2, 3, i + 1)
        _, sample_mean, confidence_interval, interval_width = \
            stat_func(sample.values)

        print(f'Значения {ci_type} для {sample.name}:'
              f'{confidence_interval}, его ширина {interval_width}')
        print(f"\t Выборочное среднее = {sample_mean}")
        sns.histplot(sample.values,
                     bins=15,
                     kde=True,
                     stat='density')
        plt.axvline(x=confidence_interval[0],
                    color='red',
                    linestyle='dashed',
                    linewidth=2,
                    label='Нижняя граница Д.И.')
        plt.axvline(x=confidence_interval[1],
                    color='green',
                    linestyle='dashed',
                    linewidth=2,
                    label='Верхняя граница Д.И.')
        plt.axvline(x=sample_mean,
                    color='orange',
                    linestyle='dashed',
                    linewidth=2,
                    label='Выборочное среднее')
        plt.title(f'{sample.name.capitalize()}')
        plt.xlabel('Значения')
        plt.ylabel('Плотность вероятности')
        plt.legend()

    plt.tight_layout()
    plt.show()
```

Изобразим доверительные интервалы для каждой выборки.

In [12]:
```
show_graph(samples,
           "асимтотический интегрвал для среднего значения",
           calc_ci_z)
```

```
Значения асимтотический интегрвал для среднего значения для маленькая выборк
а:[3.3658 5.8001], его ширина 2.4343
         Выборочное среднее = 4.583
Значения асимтотический интегрвал для среднего значения для средняя выборка:
[4.5724 5.3198], его ширина 0.7474
         Выборочное среднее = 4.9461
Значения асимтотический интегрвал для среднего значения для большая выборка:
[4.9849 5.1108], его ширина 0.1259
         Выборочное среднее = 5.0479
```



Задание №3.2

Напишем функцию для построения точного доверительного интервала для среднего значения. Точный доверительный инетервал строиться на распределении Студента.

$$\hat{x} \pm t_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{n}}$$

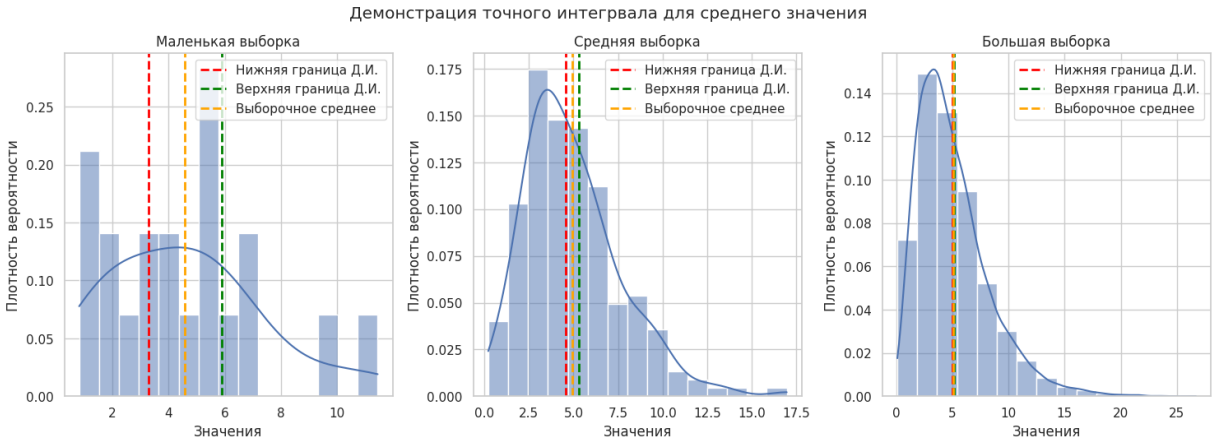
Где $z_{1-\frac{\alpha}{2}}$ - это квантиль распределения Студента для заданного уровня значимости, где количество степеней свободы равно количеству значений в выборке -1.

In [13]:
```
def calc_ci_t(sample: np.ndarray,
              alpha: float = 0.05) -> tuple:
    t_score = t(len(sample) - 1).ppf(1 - alpha / 2)

    sample_mean = np.mean(sample)
    sample_std = np.std(sample, ddof=1)

    margin_of_error = t_score * \
        (sample_std / np.sqrt(len(sample)))
    confidence_interval = (
         sample_mean - margin_of_error,
         sample_mean + margin_of_error)
    interval_width = 2 * margin_of_error
    return ("Точный доверительный интервал",
            np.round(sample_mean, 4),
            np.round(confidence_interval, 4),
            np.round(interval_width, 4))
```

Для каждой выборки выведем асимптотический доверительный интервал для среднего значения.

In [14]:
```
show_graph(samples,
           "точного интегрвала для среднего значения",
           calc_ci_t)
```

```
Значения точного интегрвала для среднего значения для маленькая выборка:[3.2
832 5.8828], его ширина 2.5996
        Выборочное среднее = 4.583
Значения точного интегрвала для среднего значения для средняя выборка:[4.570
1 5.3221], его ширина 0.752
        Выборочное среднее = 4.9461
Значения точного интегрвала для среднего значения для большая выборка:[4.984
9 5.1108], его ширина 0.1259
        Выборочное среднее = 5.0479
```



Задание №3.3

Построим эфронов доверительный интервал для среднего, медианы, моды, дисперсии. Эфроновы доверительные интервалы строятся на базе бутстрапа. Напишем функцию для вычисления доверительных интервалов для необходимых статистик. Отдельно обработаем случай для моды, будем округлять до 1 знака после запятой.

In [15]:
```python
russian_names = {'mean':'среднее',
                 'mode':'мода',
                 'median':'медиана',
                 'var':'дисперсия'}

def efron_ci_stat(sample: np.ndarray,
                  stat_func: Callable,
                  num_bootstrap_samples: int = 1_000,
                  alpha: float = 0.05) -> tuple:
    bootstrap_stats = []

    for _ in range(num_bootstrap_samples):
        bootstrap_sample = resample(sample)
        if stat_func is mode:
            bootstrap_sample = np.round(bootstrap_sample, 1)
            bootstrap_stats.append(stat_func(bootstrap_sample)[0])
        else:
            bootstrap_stats.append(stat_func(bootstrap_sample))

    lower_bound = np.percentile(bootstrap_stats, (alpha / 2) * 100)
    upper_bound = np.percentile(bootstrap_stats, (1 - alpha / 2) * 100)
```

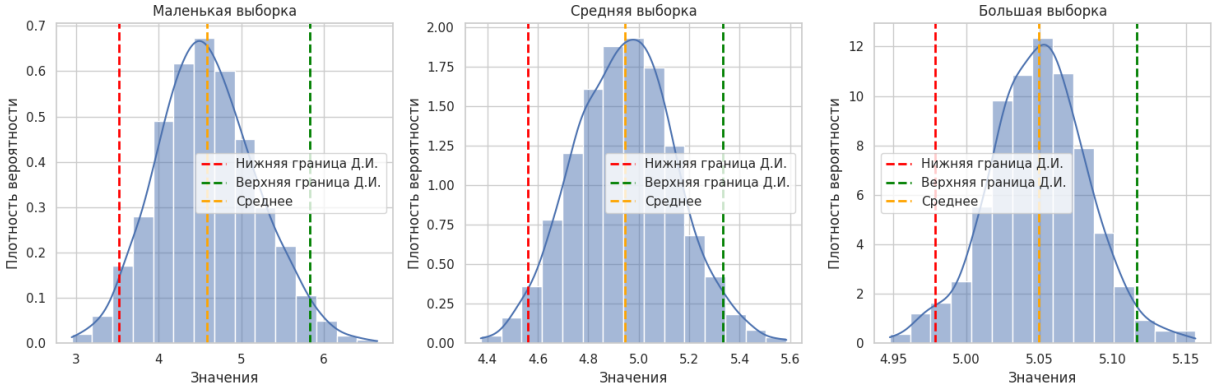
```
    confidence_interval = [lower_bound, upper_bound]
    interval_width = upper_bound - lower_bound
    margin_of_error = interval_width / 2
    return (f'Эфронов доверительный интервал для'
             '{russian_names[stat_func.__name__]}',
            np.round(bootstrap_stats, 4),
            np.round(margin_of_error, 4),
            np.round(confidence_interval, 4),
            np.round(interval_width, 4))
```

In [16]:

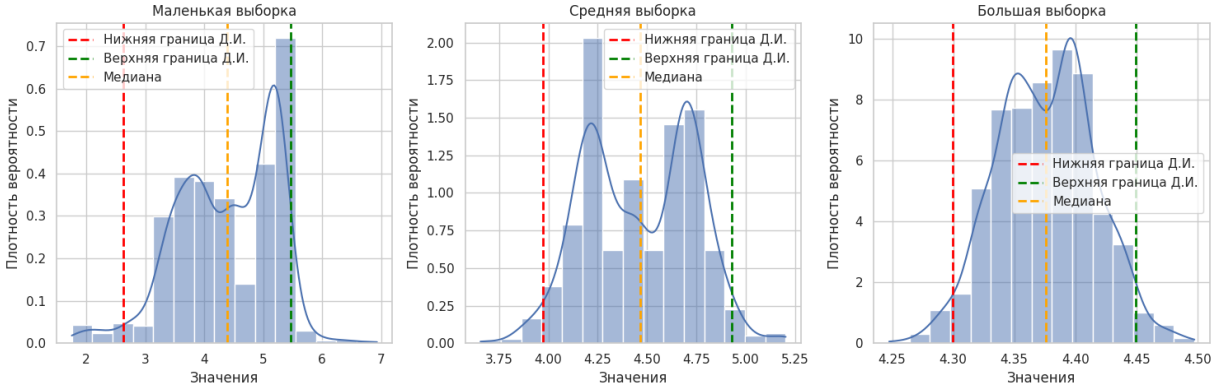
```
stat_funcs = [np.mean, np.median, mode, np.var]
for stat_func in stat_funcs:
    print(f'Демонстрация эфронова доверительного интервала для'
          f'{russian_names[stat_func.__name__]}:')
    plt.figure(figsize=(15, 5))
    for i, sample in enumerate(samples):
        plt.subplot(1, len(samples), i + 1)

        _, bootstrap_stats, __, confidence_interval, interval_width = \
        efron_ci_stat(sample.values, stat_func)

        print(f'\t Значения доверительного интервала для {sample.name}:'
              f'{confidence_interval}, его ширина {interval_width}')

        sns.histplot(bootstrap_stats,
                     bins=15,
                     kde=True,
                     stat='density')
        plt.axvline(x=confidence_interval[0],
                    color='red',
                    linestyle='dashed',
                    linewidth=2,
                    label='Нижняя граница Д.И.')
        plt.axvline(x=confidence_interval[1],
                    color='green',
                    linestyle='dashed',
                    linewidth=2,
                    label='Верхняя граница Д.И.')
        if stat_func == mode:
            res = stat_func(sample.values)[0]
        else:
            res = stat_func(sample.values)
        plt.axvline(x=np.mean(bootstrap_stats),
                    color='orange',
                    linestyle='dashed',
                    linewidth=2,
                    label=russian_names[stat_func.__name__]\
                    .capitalize())
        plt.title(f'{sample.name.capitalize()}')
        plt.xlabel('Значения')
        plt.ylabel('Плотность вероятности')
        plt.legend()

    plt.tight_layout()
```

```
    plt.show()
    print()
```

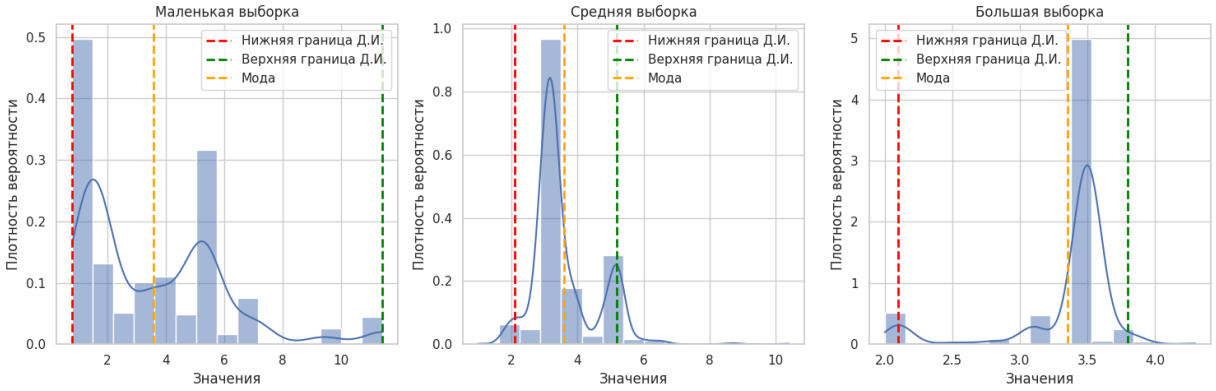
```
Демонстрация эфронова доверительного интервала длясреднее:
        Значения доверительного интервала для маленькая выборка:[3.5235 5.8
292], его ширина 2.3057
        Значения доверительного интервала для средняя выборка:[4.5601 5.332
7], его ширина 0.7726
        Значения доверительного интервала для большая выборка:[4.9785 5.116
4], его ширина 0.1379
```



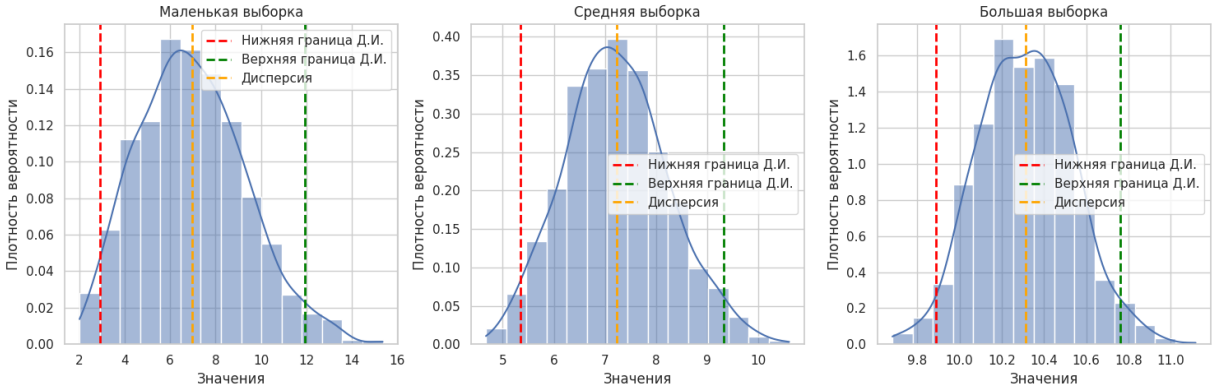

```
Демонстрация эфронова доверительного интервала длямедиана:
        Значения доверительного интервала для маленькая выборка:[2.6264 5.4
656], его ширина 2.8391
        Значения доверительного интервала для средняя выборка:[3.9734 4.929
5], его ширина 0.9561
        Значения доверительного интервала для большая выборка:[4.2999 4.449
7], его ширина 0.1497
```




```
Демонстрация эфронова доверительного интервала длямода:
        Значения доверительного интервала для маленькая выборка:[ 0.8 11.
4], его ширина 10.6
        Значения доверительного интервала для средняя выборка:[2.1 5.2], ег
о ширина 3.1
        Значения доверительного интервала для большая выборка:[2.1 3.8], ег
о ширина 1.7
```




```
Демонстрация эфронова доверительного интервала длядисперсия:
        Значения доверительного интервала для маленькая выборка:[ 2.9037 1
1.9432], его ширина 9.0395
        Значения доверительного интервала для средняя выборка:[5.348  9.321
4], его ширина 3.9734
        Значения доверительного интервала для большая выборка:[ 9.8866 10.7
625], его ширина 0.8759
```



Итог для 3-тьего задания

Выведем сводную таблицу для построения выводов.

In [17]:
```
data = []

for sample in samples:
    ci_z = calc_ci_z(sample.values)
    name, _, confidence_interval, interval_width = \
        calc_ci_z(sample.values)
    data.append({'sample_name': sample.name.capitalize(),
                 'stat_func': 'среднее'.capitalize(),
                 'method_name': name,
                 'confidence_interval': confidence_interval,
                 'interval_width': interval_width})

    ci_t = calc_ci_t(sample.values)
    name, _, confidence_interval, interval_width = \
        calc_ci_t(sample.values)
    data.append({'sample_name': sample.name.capitalize(),
                 'stat_func': 'среднее'.capitalize(),
                 'method_name': name,
                 'confidence_interval': confidence_interval,
```

```
                    'interval_width': interval_width})

    for stat_func in stat_funcs:
        name, bootstrap_stats, _, confidence_interval, interval_width = \
            efron_ci_stat(sample.values, stat_func)
        data.append({'sample_name': sample.name.capitalize(),
                    'stat_func': russian_names[stat_func.__name__].capitali
                    'method_name': ' '.join(name.split()[:-2]),
                    'confidence_interval': confidence_interval,
                    'interval_width': interval_width})
for line in data:
    true_val = 5
    if line['stat_func'] == 'дисперсия':
        true_val = 10
    if line['stat_func'] == 'Медиана':
        true_val = 4.3
    if line['stat_func'] == 'Мода':
        true_val = 3
    line['true_val'] = true_val


df = pd.DataFrame(data).sort_values(['sample_name',
                                     'stat_func',
                                     'method_name']).\
    reset_index(drop=True)
df.rename(columns = {'sample_name': 'Название выборки',
                    'stat_func': 'Статистика',
                    'method_name': 'Название метода',
                    'confidence_interval': 'Д. И.',
                    'interval_width': 'Ширина Д. И.',
                    'true_val': 'Статистика из Г. С.'})
```

Out[17]:

	Название выборки	Статистика	Название метода	Д. И.	Ширина Д. И.	Статистика из Г. С.
0	Большая выборка	Дисперсия	Эфронов доверительный	[9.9134, 10.7489]	0.8355	5.0
1	Большая выборка	Медиана	Эфронов доверительный	[4.3087, 4.4491]	0.1404	4.3
2	Большая выборка	Мода	Эфронов доверительный	[2.1, 3.8]	1.7000	3.0
3	Большая выборка	Среднее	Асимтотический доверительный интервал	[4.9849, 5.1108]	0.1259	5.0
4	Большая выборка	Среднее	Точный доверительный интервал	[4.9849, 5.1108]	0.1259	5.0
5	Большая выборка	Среднее	Эфронов доверительный	[4.9819, 5.1119]	0.1300	5.0
6	Маленькая выборка	Дисперсия	Эфронов доверительный	[3.0036, 11.7016]	8.6980	5.0
7	Маленькая выборка	Медиана	Эфронов доверительный	[2.7503, 5.4656]	2.7152	4.3
8	Маленькая выборка	Мода	Эфронов доверительный	[0.8, 11.4]	10.6000	3.0
9	Маленькая выборка	Среднее	Асимтотический доверительный интервал	[3.3658, 5.8001]	2.4343	5.0
10	Маленькая выборка	Среднее	Точный доверительный интервал	[3.2832, 5.8828]	2.5996	5.0
11	Маленькая выборка	Среднее	Эфронов доверительный	[3.3995, 5.8098]	2.4102	5.0
12	Средняя выборка	Дисперсия	Эфронов доверительный	[5.5386, 9.2779]	3.7393	5.0
13	Средняя выборка	Медиана	Эфронов доверительный	[3.9382, 4.9295]	0.9914	4.3
14	Средняя выборка	Мода	Эфронов доверительный	[2.1, 6.4]	4.3000	3.0
15	Средняя выборка	Среднее	Асимтотический доверительный интервал	[4.5724, 5.3198]	0.7474	5.0
16	Средняя выборка	Среднее	Точный доверительный интервал	[4.5701, 5.3221]	0.7520	5.0
17	Средняя выборка	Среднее	Эфронов доверительный	[4.5976, 5.3248]	0.7273	5.0

Интерпритируем результаты

Эфронов доверительный интервал

Первое, что бросается в глаза - это большая ширина для моды, он в разы отличается от доверительных интервалов для других характеристик. Связано это с тем, что мода очень вариативная характеристика, в отличие, к примеру, от среднего (стоит отметить, что вычисление моды производилось с округлением, что "дискретизирует" непрерывную величину). Так же большой доверительный интервал имеет дисперсия, точно по этой же причине.

Точный доверительный интервал

Точный доверительный интервал, в сравнении с остальными методами построения доверительных интервалов, имеет немного большую ширину, связано это с тем, что этот метод учитывает количество данных в выборке.

Асимптотический доверительный интервал

Тут сказать нечего.

Итог

Мы видим, что все доверительные интервалы включают в себя значения статистик из генеральной совокупности, кроме того, мы обратную зависимость количества значений в выборке и ширины доверительного интервала.