Санкт-Петербургский Политехнический
Университет им. Петра Великого

Институт прикладной математики и механики

Кафедра прикладной математики

Лабораторная работа №2
Характеристики положения выборки

3 курс, группа 33631/2

Студент Д. А. Плаксин

Преподаватель Баженов А. Н.

Санкт-Петербург
2019 г.

Содержание

1 Список таблиц

2 Постановка задачи

Любыми средствами сгенерировать выборки размеров 20, 60, 100 элементов для 5ти распределений [2]. Для каждой выборки вычислить $\overline{x}$, $med\ x$, Z_R, Z_Q, Z_{tr}, при $r = \frac{n}{4}$.

Распределения:

1. Стандартное нормальное распределение
2. Стандартное распределение Коши
3. Распределение Лапласа с коэффициентом масштаба $\sqrt{2}$ и нулевым коэффициентом сдвига.
4. Равномерное распределение на отрезке $\left[-\sqrt{3}, \sqrt{3}\right]$
5. Распределение Пуассона со значением матожидания равным двум.

3 Теория

1. Выборочное среднее [3]:
$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$
2. Выборочная медиана [4]:
$$med\ x = \begin{cases} x_{k+1}, & n = 2k+1 \\ \frac{1}{2}\left(x_k + x_{k+1}\right), & n = 2k \end{cases} \tag{2}$$
3. Полусумма экстремальных значений [5]:
$$Z_R = \frac{1}{2}\left(x_1 + x_n\right) \tag{3}$$
4. Полусумма квартилей [6]:
$$Z_Q = \frac{1}{2}\left(Z_{\frac{1}{4}} + Z_{\frac{3}{4}}\right) \tag{4}$$
5. Усечённое среднее [7]:
$$Z_{tr} = \frac{1}{n-2r}\sum_{i=r+1}^{n-r} x_i \tag{5}$$

4 Реализация

Для генерации выборки был использован *Python* 3.7: модуль *random* библиотеки *numpy* [1] для генерации случайных чисел с различными распределениями. Также с помощью библиотеки *numpy* были вычислены характеристики положения.

После вычисления характеристик положения 1000 раз находится среднее значение и дисперсия:
$$E(z) = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{6}$$
$$D(z) = E\left(z^2\right) - E^2(z) \tag{7}$$

5 Результаты

Таблица 1: Стандартное нормальное распределение.

$n = 20$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	−0.00	−0.00	0.0	0.00	−0.01
$D =$	0.051763	0.072023	0.149073	0.058764	0.059323
$n = 60$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.00	0.00	−0.0	−0.00	−0.00
$D =$	0.016322	0.026125	0.110309	0.020489	0.020389
$n = 100$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.000	−0.00	−0.00	−0.00	−0.00
$D =$	0.009563	0.015479	0.096256	0.012569	0.012224

Таблица 2: Стандартное распределение Коши.

$n = 20$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	3.414266	−0.012522	−5.262327	−0.027313	0.020763
$D =$	12977.113689	0.130212	12626.531931	0.371684	0.150542
$n = 60$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	−1.180135	−0.008143	−69.304468	0.007723	−0.006201
$D =$	1424.446598	0.038886	6240734.028690	0.086309	0.042207
$n = 100$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.811064	−0.003271	30.434787	0.002690	−0.008544
$D =$	350.070803	0.023004	1465401.042218	0.054476	0.025649

Таблица 3: Распределение Лапласа.

$n = 20$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	−0.01	−0.01	−0.0	0.00	−0.00
$D =$	0.047943	0.031492	0.436478	0.047849	0.031600
$n = 60$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.00	−0.00	−0.0	−0.00	−0.00
$D =$	0.017707	0.010050	0.455793	0.014958	0.010114
$n = 100$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.005	0.004	0.0	0.00	−0.000
$D =$	0.009733	0.006307	0.409248	0.010233	0.006262

Таблица 4: Равномерное распределение.

$n = 20$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	−0.00	−0.0	0.00	−0.00	0.00
$D =$	0.046929	0.126544	0.013805	0.070389	0.098176
$n = 60$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.00	−0.00	−0.000	0.00	0.00
$D =$	0.018149	0.048531	0.001583	0.023893	0.032310
$n = 100$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	0.00	0.00	0.0010	0.00	0.00
$D =$	0.010555	0.029134	0.000562	0.014285	0.020388

Таблица 5: Распределение Пуассона.

$n = 20$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	1.99	1.8	2.5	1.9	1.8
$D =$	0.096696	0.201928	0.325744	0.126251	0.125845
$n = 60$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	1.99	1.92	2.9	1.94	1.84
$D =$	0.032178	0.063700	0.245694	0.033469	0.046742
$n = 100$	average	med	Z_R	Z_Q	$Z_{tr},\ r = \frac{n}{4}$
$E =$	1.99	1.96	3.1	1.96	1.83
$D =$	0.020435	0.030204	0.239758	0.018016	0.026728

6 Обсуждение

При вычислении средних значений пришлось отбрасывать некоторое число знаков после запятой, так как получаемая дисперсия не могла гарантировать получаемое точное значение.

Иными словами дисперсия может гарантировать порядок точности среднего значения только до первого значащего знака после запятой в дисперсии включительно.

Единственным исключением [в отбрасывании знаков после запятой] стало стандартное распределение Коши, так как оно имеет бесконечную дисперсию, а значит может гарантировать сколь угодно большую точность.

7 Выводы

В процессе работы вычислены значения характеристик положения для определённых распределений на выборках фиксированной мощности и получено следующее ранжирование характеристик положения:

1. Стандартное нормальное распределение

$$\overline{x} < Z_{tr} < Z_Q < med\, x < Z_R$$

2. Стандартное распределение Коши

$$med\, x < Z_Q < Z_{tr} < \overline{x} < Z_R$$

3. Распределение Лапласа (коэффициент масштаба $\sqrt{2}$ коэффициент сдвига равен нулю)

$$med\, x < Z_{tr} < \overline{x} < Z_Q < Z_R$$

4. Равномерное распределение на отрезке $\left[-\sqrt{3}, \sqrt{3}\right]$

$$Z_R < \overline{x} < Z_{tr} < Z_Q < med\, x$$

5. Распределение Пуассона (значение мат ожидания равно 3)

$$\overline{x} < Z_{tr} < Z_Q < med\, x < Z_R$$

8 Список литературы

[1] Модуль numpy - https://physics.susu.ru/vorontsov/language/numpy.html

[2] Формулы распределений - https://vk.com/doc184549949_491827451

[3] Выборочное среднее - https://en.wikipedia.org/wiki/Sample_mean_and_covariance

[4] Выборочная медиана - http://femto.com.ua/articles/part_1/2194.html

[5] Полусумма экстремальных значений - https://studopedia.info/8-56888.html

[6] Квартили - https://studfiles.net/preview/2438125/page:13/

[7] Усечённое среднее - https://ole-olesko.livejournal.com/15773.html

9 Приложения

Код отчёта: https://github.com/MisterProper9000/MatStatLabs/blob/master/MatStatLab2/MatStatLab2.tex

Код лаборатрной: https://github.com/MisterProper9000/MatStatLabs/blob/master/MatStatLab2/MatStatLab2.py

```

import numpy
import sys


POISSON_PARAM = 2
UNIFORM_LEFT = -numpy.sqrt(3)
UNIFORM_RIGHT = numpy.sqrt(3)
LAPLAS_COEF = numpy.sqrt(2)
selection = [20, 60, 100]
selection = numpy.sort(selection)

def standart_normal(x):
    return (1 / numpy.sqrt(2*numpy.pi)) * numpy.exp(- x * x / 2)


def standart_cauchy(x):
    return 1 / (numpy.pi * (1 + x*x))


def laplace(x):
    return 1 / LAPLAS_COEF * numpy.exp (-LAPLAS_COEF * numpy.abs(x))


```

```python
def uniform(x):
    flag2 = x <= UNIFORM_RIGHT
    flag1 = x >= UNIFORM_LEFT
    return 1 / (UNIFORM_RIGHT - UNIFORM_LEFT) * flag1 * flag2


def poisson(x):
    k = POISSON_PARAM
    return (numpy.power(x, k) / numpy.math.factorial(k)) * numpy.exp(-x)


func_dict = {
    'normal': standart_normal,
    'cauchy': standart_cauchy,
    'laplace': laplace,
    'uniform': uniform,
    'poisson': poisson
}


def generate_laplace(x):
    return numpy.random.laplace(0, 1/LAPLAS_COEF, x)


def generate_uniform(x):
    return numpy.random.uniform(UNIFORM_LEFT, UNIFORM_RIGHT, x)


def generate_poisson(x):
    return numpy.random.poisson(POISSON_PARAM, x)


generate_dict = {
    'normal': numpy.random.standard_normal,
    'cauchy': numpy.random.standard_cauchy,
    'laplace': generate_laplace,
    'uniform': generate_uniform,
    'poisson': generate_poisson
}


def Zr(x):
    return (numpy.amin(x) + numpy.amax(x))/2


def Zq(x):
    return (numpy.quantile(x, 1/4) + numpy.quantile(x, 3/4) ) /2


def Ztr(x):
    length = x.size
    r = (int)(length / 4)
    sum = 0
    for i in range(r, length - r):
        sum += x[i]
    return sum/(length - 2 * r)


pos_characteristic_dict = {
    'average': numpy.mean,
    'med': numpy.median,
    'Zr': Zr,
    'Zq': Zq,
    'Ztr r = n/4': Ztr
}

```

```
pos_char_name = [
    'average',
    'med',
    'Zr',
    'Zq',
    'Ztr r = n/4'
]


def E(z):
    return numpy.mean(z)


def D(z):
    return numpy.var(z)



f = open('out.csv', 'w')
sys.stdout = f


def research(dist_type):
    print('----------------------------------------')
    print(dist_type)
    for num in selection:
        #print(num)
        print_table = {
            'E': [],
            'D': []
        }
        for pos_name in pos_char_name:
            z = []
            for i in range(0, 1000):
                arr = numpy.sort(generate_dict[dist_type](num))
                z.append(pos_characteristic_dict[pos_name](arr))
            print_table['E'].append(E(z))
            print_table['D'].append(D(z))

        print()
        print("%-10s;" %('n = %i' %num), end="")
        for pos_name in pos_char_name:
            print("%-12s;" % pos_name, end="")

        print()
        print("%-10s;" % ('E ='), end="")
        for e in print_table['E']:
            print("%-12f;" % e, end="")

        print()
        print("%-10s;" %('D ='), end="")
        for d in print_table['D']:
            print("%-12f;" % d, end="")
        print()




research('normal')
research('cauchy')
research('laplace')
research('uniform')
research('poisson')
```