

Федеральное государственное автономное образовательное
учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Отчёт

По практической работе №6

по дисциплине «Теория вероятностей»

Вариант: 12

Работу выполнил:

Поленов Кирилл Александрович

Группа Р3213

Работу принял:

Селина Елена Георгиевна

г. Санкт-Петербург 2024

Оглавление

Задание.....	3
Исходный код программы	3
UML диаграмма проекта.....	Ошибка! Закладка не определена.
Выводы.....	Ошибка! Закладка не определена.

Задание

Каждый студент получает выборку из 20 чисел. Необходимо определить следующие статистические характеристики: вариационный ряд, экстремальные значения и размах, оценки математического ожидания и среднеквадратического отклонения эмпирическую функцию распределения и её график, гистограмму и полигон приведенных частот группированной выборки. Для расчета характеристик и построения графиков нужно написать программу на одном из языков программирования. Листинг программы и результаты работы должны быть представлены в отчете по практической работе. Стандартные функции статистики использовать нельзя.

12	0.41	-1.53	0.85	1.54	1.24	1.08	-0.92	1.15	0.26	1.57
	1.63	-0.20	0.09	0.25	-0.26	0.42	-0.91	-0.82	0.96	0.72

Исходный код программы

```
from math import sqrt, log2, ceil
import matplotlib.pyplot as mathplot

data = [0.41, 1.63, -1.53, -0.20, 0.85, 0.09, 1.54, 0.25, 1.24, -0.26,
        1.08, 0.42, -0.92, -0.91, 1.15, -0.82, 0.26, 0.96, 1.57, 0.72]

n = len(data)

# Вариационный ряд
var_array = sorted(data)
# Статистический ряд
elements = set(var_array)
freqs = [var_array.count(i) for i in elements]

#min
x_min = min(var_array)
#max
x_max = max(var_array)
# Размах
range_val = x_max - x_min
# Мат. ожидание
math_exp = round(sum(data)/n, 2)
# Дисперсия
variance = round(sum((x - math_exp)**2 for x in data)/n, 2)
# Исправленная дисперсия
corr_variance = round((n/(n-1))*variance, 2)
# Среднеквадратичное отклонение
avg_deviance = round(sqrt(variance), 2)
# Исправление ср-кв. отклонение
corr_deviance = round(sqrt(corr_variance), 2)
# Эмпирическая функция распределения
emp_func = [(var_array[i], round((i+1)/n, 2)) for i in range(len(var_array))]
emp_func.insert(0, (x_min, 0.0))
emp_func.append((x_max, 1.0))
# Кол-во интервалов
intervals_count = ceil(1+log2(n))
# Длина интервала
interval_length = round((x_max-x_min)/intervals_count, 2)
# Начало интервала
```

```

interval_begin = round((x_min - interval_length/2), 2)
# Интервальное распределение частот
intervals = [(round(interval_begin + i * interval_length, 2),
              round(interval_begin + (i + 1) * interval_length, 2))
              for i in range(intervals_count)] 

interval_freqs = [sum(1 for x in data if interval[0] <= x < interval[1])
                  for interval in intervals]

interval_probs = [round(f/n, 2) for f in interval_freqs]
# Медиана
def median():
    mid = n // 2
    if n % 2 == 0:
        # если чет кол-во элементов, вернуть среднее двух по центру
        return (var_array[mid-1] + var_array[mid]) / 2
    else:
        # иначе берем элемент со среднего индекса
        return var_array[mid]

# Нахождение моды
def mode():
    freq = {}
    for num in var_array:
        freq[num] = freq.get(num, 0) + 1 # Подсчет частоту каждого числа

    max_freq = max(freq.values()) # Максимальная частота
    modes = [num for num, count in freq.items() if count == max_freq]

    if len(modes) == 1:
        # если мода одна, её и возвращаем
        return modes[0]
    else:
        # иначе (несколько значений встречаются одинаково)
        return None

# Вывод в консоль
print("Вариационный ряд:", var_array, "\n")
print("Статистический ряд:")
print("\tЭлементы:", elements)
print("\tЧастоты:", freqs, "\n")
print(f"Минимальное значение: {x_min},\nМаксимальное значение:
{x_max},\nРазмах: {range_val}\n")
print(f"Математическое ожидание: {math_exp}")
print(f"Дисперсия: {variance}")
print(f"Исправленная дисперсия: {corr_variance}")
print(f"Среднеквадратическое отклонение: {avg_deviance}")
print(f"Исправленное среднеквадратическое отклонение: {corr_deviance}")
print(f"Медиана: {median()}")
mode = mode()
print(f"Мода: {mode if mode is not None else 'Мода отсутствует'}\n")

print("Эмпирическая функция распределения:")
print(f"\tПри x <= {x_min}: 0")
for i in range(1, len(emp_func) - 1):
    print(f"\tПри {emp_func[i-1][0]} < x <= {emp_func[i][0]} :
{emp_func[i][1]}")
print(f"\tПри x > {x_max}: 1.0")

print(f"Количество интервалов: {intervals_count},"
      f" Длина интервала: {interval_length}, Начало первого интервала:
{intervals_count}\n")

```

```

print("Интервальное распределение частот:")
for i, interval in enumerate(intervals):
    print(f"\t[{interval[0]:.2f}, {interval[1]:.2f}):\n"
          f" Частота - {interval_freqs[i]}, Частотность - "
          f"{interval_probs[i]}")

# Графики
# Гистограмма
mathplot.bar([interval[0] for interval in intervals], interval_probs,
            width=interval_length, alpha=0.7, edgecolor='black', align='edge')
mathplot.title("Гистограмма частотностей")
mathplot.xlabel("Интервалы")
mathplot.ylabel("Частотности")
mathplot.grid()
mathplot.show()

# Полигон частотностей
x = [interval[0] + interval_length / 2 for interval in intervals]
y = interval_probs
mathplot.plot(x, y, marker='o', color='b')
mathplot.title("Полигон частотностей")
mathplot.xlabel("Интервалы")
mathplot.ylabel("Частотности")
mathplot.grid()
mathplot.show()

# График эмпирической функции распределения
emp_x = [x[0] for x in emp_func]
emp_y = [x[1] for x in emp_func]
mathplot.step(emp_x, emp_y, where="post", color="r", label="Эмпирическая
функция")
mathplot.title("График эмпирической функции распределения")
mathplot.xlabel("x")
mathplot.ylabel("F(x)")
mathplot.grid()
mathplot.legend()
mathplot.show()

```

Результаты выполнения программы

Вариационный ряд: [-1.53, -0.92, -0.91, -0.82, -0.26, -0.2, 0.09, 0.25, 0.26, 0.41, 0.42, 0.72, 0.85, 0.96, 1.08, 1.15, 1.24, 1.54, 1.57, 1.63]

Статистический ряд:

Элементы: {-0.92, -0.91, -0.26, -0.82, -0.2, 0.25, 0.26, 0.41, 0.42, 0.96, 1.08, 0.85, 0.09, 1.54, 1.63, 0.72, 1.15, 1.24, 1.57, -1.53}

Частоты: [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

Минимальное значение: -1.53,

Максимальное значение: 1.63,

Размах: 3.16

Математическое ожидание: 0.38

Дисперсия: 0.81

Исправленная дисперсия: 0.85

Среднеквадратическое отклонение: 0.9

Исправленное среднеквадратическое отклонение: 0.92

Медиана: 0.415

Мода: Мода отсутствует

Эмпирическая функция распределения:

При $x \leq -1.53$: 0

При $-1.53 < x \leq -1.53$: 0.05

При $-1.53 < x \leq -0.92$: 0.1

При $-0.92 < x \leq -0.91$: 0.15

При $-0.91 < x \leq -0.82$: 0.2

При $-0.82 < x \leq -0.26$: 0.25

При $-0.26 < x \leq -0.2$: 0.3

При $-0.2 < x \leq 0.09$: 0.35

При $0.09 < x \leq 0.25$: 0.4

При $0.25 < x \leq 0.26$: 0.45

При $0.26 < x \leq 0.41$: 0.5

При $0.41 < x \leq 0.42$: 0.55

При $0.42 < x \leq 0.72$: 0.6

При $0.72 < x \leq 0.85$: 0.65

При $0.85 < x \leq 0.96$: 0.7

При $0.96 < x \leq 1.08$: 0.75

При $1.08 < x \leq 1.15$: 0.8

При $1.15 < x \leq 1.24$: 0.85

При $1.24 < x \leq 1.54$: 0.9

При $1.54 < x \leq 1.57$: 0.95

При $1.57 < x \leq 1.63$: 1.0

При $x > 1.63$: 1.0

Количество интервалов: 6, Длина интервала: 0.53, Начало первого интервала: 6

Интервальное распределение частот:

$[-1.79, -1.26]$: Частота - 1, Частотность - 0.05

$[-1.26, -0.73]$: Частота - 3, Частотность - 0.15

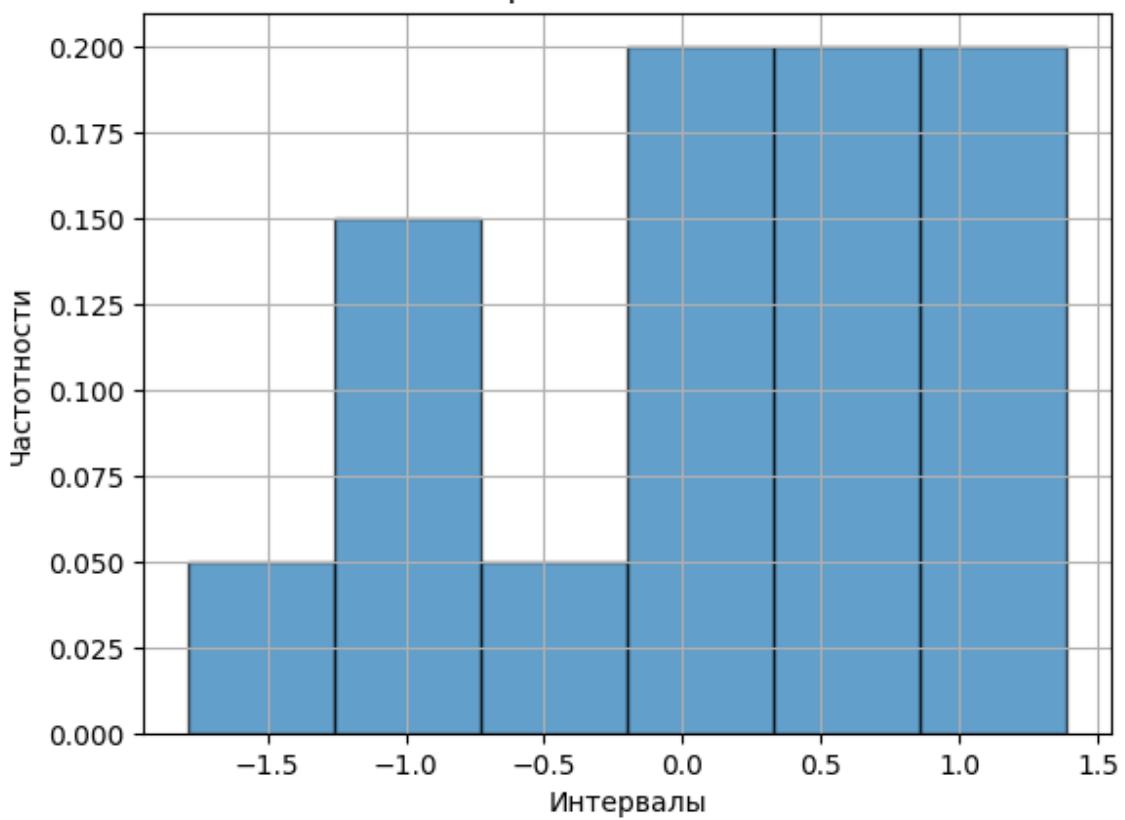
$[-0.73, -0.20]$: Частота - 1, Частотность - 0.05

$[-0.20, 0.33]$: Частота - 4, Частотность - 0.2

$[0.33, 0.86]$: Частота - 4, Частотность - 0.2

$[0.86, 1.39]$: Частота - 4, Частотность - 0.2

Гистограмма частотностей



Полигон частотностей

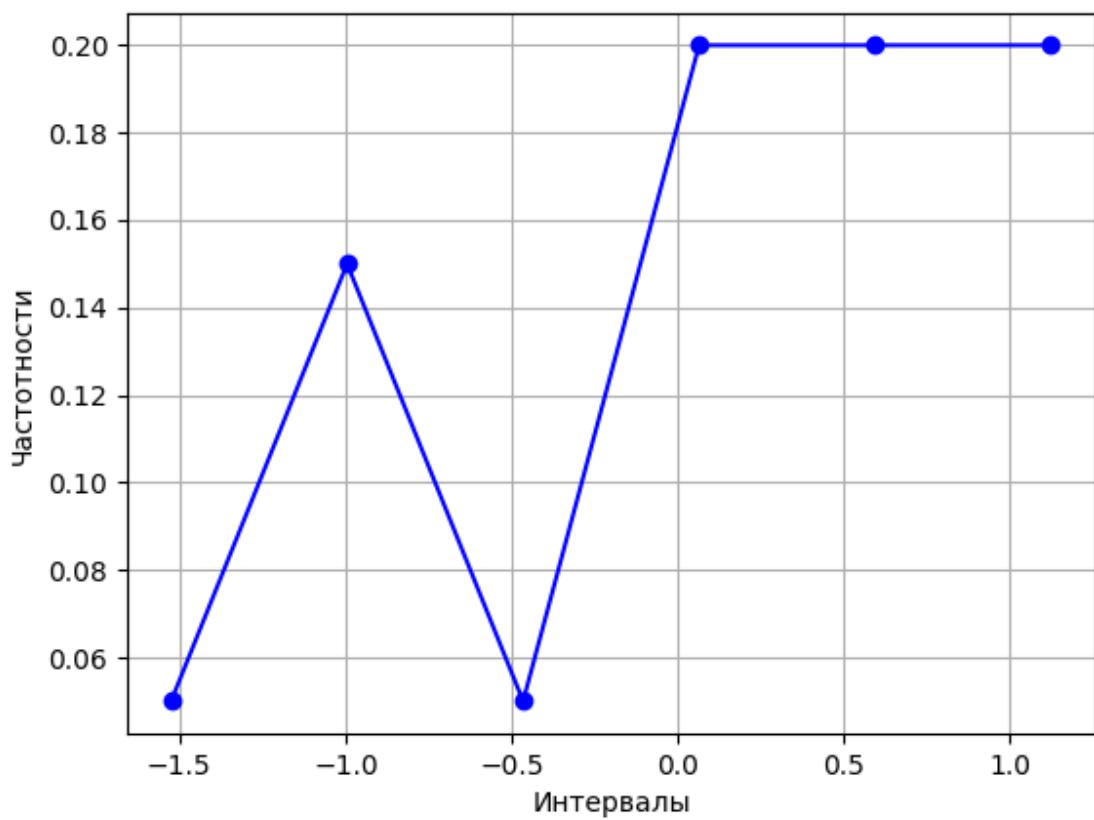


График эмпирической функции распределения

